SIEMENS

System-Bausatz SIKIT-DK/8080 Bauanleitung

System SAB 8080

VX

blank page

Schuhl #20249

Inhaltsverzeichnis

Einführung

Zusammenbau

Erklärungen zur Arbeitsweise Ihres Mikrocomputers

Endzusammenbau und Überprüfung

Das MONITOR-Programm

Ausdruck des MONITOR-Primärprogrammes

Gesamtschaltung des SIKIT-DK/8080-Computers

Gesamtbestückungsplan

Anschriften unserer Geschäftsstellen

Herausgegeben von
Siemens AG, Bereich Bauelemente, Balanstraße 73, 8000 München 80.

Ursprungsfassung in englisch: © Intel Corporation, USA.

Für die angegebenen Schaltungen, Beschreibungen und Tabellen wird keine Gewähr bezüglich der Freiheit von Rechten Dritter übernommen.

Liefermöglichkeiten und technische Änderungen vorbehalten.

Fragen über Technik, Preise und Liefermöglichkeiten richten Sie bitte an unsere Zweigniederlassungen im Inland, Abteilung VB oder an unsere Landesgesellschaften im Ausland (siehe Geschäftsstellenverzeichnis).

SIEMENS

SAB 8080
System-Bausatz SIKIT-DK/8080

Bauanleitung

Ausgabe 1976

SIEMENS AKTIENGESELLSCHAFT

VX

JUMP

C3	JMP	Adr
C2	JNZ	Adr
CA	JZ	Adr
D2	JNC	Adr
DA	JC	Adr
E2	JPO	Adr
EA	JPE	Adr
F2	JP	Adr
FA	JM	Adr
E9	PCHL	

CALL

CD	CALL	Adr
C4	CNZ	Adr
CC	CZ	Adr
D4	CNC	Adr
DC	CC	Adr
E4	CPO	Adr
EC	CPE	Adr
F4	CP	Adr
FC	CM	Adr

RETURN

C9	RET
C0	RNZ
C8	RZ
D0	RNC
D8	RC
E0	RPO
E8	RPE
F0	RP
F8	RM

RESTART

C7	RST	0
CF	RST	1
D7	RST	2
DF	RST	3
E7	RST	4
EF	RST	5
F7	RST	6
FF	RST	7

ROTATE

07	RLC
0F	RRC
17	RAL
1F	RAR

CONTROL

00	NOP
76	HLT
F3	DI
FB	EI

MOVE

40	MOV	B,B
41	MOV	B,C
42	MOV	B,D
43	MOV	B,E
44	MOV	B,H
45	MOV	B,L
46	MOV	B,M
47	MOV	B,A
48	MOV	C,B
49	MOV	C,C
4A	MOV	C,D
4B	MOV	C,E
4C	MOV	C,H
4D	MOV	C,L
4E	MOV	C,M
4F	MOV	C,A
50	MOV	D,B
51	MOV	D,C
52	MOV	D,D
53	MOV	D,E
54	MOV	D,H
55	MOV	D,L
56	MOV	D,M
57	MOV	D,A

MOVE (cont)

58	MOV	E,B
59	MOV	E,C
5A	MOV	E,D
5B	MOV	E,E
5C	MOV	E,H
5D	MOV	E,L
5E	MOV	E,M
5F	MOV	E,A
60	MOV	H,B
61	MOV	H,C
62	MOV	H,D
63	MOV	H,E
64	MOV	H,H
65	MOV	H,L
66	MOV	H,M
67	MOV	H,A
68	MOV	L,B
69	MOV	L,C
6A	MOV	L,D
6B	MOV	L,E
6C	MOV	L,H
6D	MOV	L,L
6E	MOV	L,M
6F	MOV	L,A
70	MOV	M,B
71	MOV	M,C
72	MOV	M,D
73	MOV	M,E
74	MOV	M,H
75	MOV	M,L

77	MOV	M,A
78	MOV	A,B
79	MOV	A,C
7A	MOV	A,D
7B	MOV	A,E
7C	MOV	A,H
7D	MOV	A,L
7E	MOV	A,M
7F	MOV	A,A

ACCUMULATOR

80	ADD	B
81	ADD	C
82	ADD	D
83	ADD	E
84	ADD	H
85	ADD	L
86	ADD	M
87	ADD	A
88	ADC	B
89	ADC	C
8A	ADC	D
8B	ADC	E
8C	ADC	H
8D	ADC	L
8E	ADC	M
8F	ADC	A
90	SUB	B
91	SUB	C
92	SUB	D
93	SUB	E
94	SUB	H
95	SUB	L
96	SUB	M
97	SUB	A
98	SBB	B
99	SBB	C
9A	SBB	D
9B	SBB	E
9C	SBB	H
9D	SBB	L
9E	SBB	M
9F	SBB	A
A0	ANA	B
A1	ANA	C
A2	ANA	D
A3	ANA	E
A4	ANA	H
A5	ANA	L
A6	ANA	M
A7	ANA	A
A8	XRA	B
A9	XRA	C
AA	XRA	D
AB	XRA	E
AC	XRA	H
AD	XRA	L
AE	XRA	M
AF	XRA	A
B0	ORA	B
B1	ORA	C
B2	ORA	D
B3	ORA	E
B4	ORA	H
B5	ORA	L
B6	ORA	M
B7	ORA	A
B8	CMP	B
B9	CMP	C
BA	CMP	D
BB	CMP	E
BC	CMP	H
BD	CMP	L
BE	CMP	M
BF	CMP	A

CONSTANT DEFINITION

0BDH, 1AH	Hex
105D, 105	Decimal
72Q, 72O	Octal
11011B, 00110B	Binary
'TEST', 'A', 'B'	ASCII

OPERATORS

(,)
* , / , MOD, SHL, SHR
+ , -
NOT
AND
OR, XOR

MOVE IMMEDIATE

06	MVI	B
0E	MVI	C
16	MVI	D
1E	MVI	E
26	MVI	H
2E	MVI	L
36	MVI	M
3E	MVI	A

D8

Acc IMMEDIATE

C6	ADI	D8
CE	ACI	D8
D6	SUI	D8
DE	SBI	D8
E6	ANI	D8
EE	XRI	D8
F6	ORI	D8
FE	CPI	D8

LOAD IMMEDIATE

01	LXI	B, D16
11	LXI	D, D16
21	LXI	H, D16
31	LXI	SP, D16

DOUBLE ADD

09	DAD	B
19	DAD	D
29	DAD	H
39	DAD	SP

INCREMENT

04	INR	B
0C	INR	C
14	INR	D
1C	INR	E
24	INR	H
2C	INR	L
34	INR	M
3C	INR	A
03	INX	B
13	INX	D
23	INX	H
33	INX	SP

DECREMENT

05	DCR	B
0D	DCR	C
15	DCR	D
1D	DCR	E
25	DCR	H
2D	DCR	L
35	DCR	M
3D	DCR	A
0B	DCX	B
1B	DCX	D
2B	DCX	H
3B	DCX	SP

LOAD/STORE

0A	LDAX	B
1A	LDAX	D
2A	LHLD	Adr
3A	LDA	Adr
02	STAX	B
12	STAX	D
22	SHLD	Adr
32	STA	Adr

STACK OPS

C5	PUSH	B
D5	PUSH	D
E5	PUSH	H
F5	PUSH	PSW
C1	POP	B
D1	POP	D
E1	POP	H
F1	POP	PSW*
E3	XTHL	
F9	SPHL	

SPECIALS

EB	XCHG	
27	DAA*	
2F	CMA	
37	STC†	
3F	CMC†	

INPUT OUTPUT

D3	OUT	D8
DB	IN	D8

PSEUDO INSTRUCTION

ORG	Adr
END	
EQU	D16
SET	D16
DS	D16
DB	D8 [ ]
DW	D16 [ ]
IF	D16
ENDIF	
MACRO	[ ]
ENDM	

STANDARD SETS

A	SET	7
B	SET	0
C	SET	1
D	SET	2
E	SET	3
H	SET	4
L	SET	5
M	SET	6
SP	SET	6
PSW	SET	6

FLAG BYTE STACK FORMAT

7	6	5	4	3	2	1	Ø
S	Z	Ø	AC	Ø	P	1	C

D8 = constant, or logical/arithmetic expression that evaluates to an 8 bit data quantity

* all Flags (C, Z, S, P, AC) affected

D16 = constant, or logical/arithmetic expression that evaluates to a 16 bit data quantity

† only CARRY affected

Adr 16 bit address

** all Flags except CARRY affected (exception INX & DCX affect no Flags)

Inhaltsverzeichnis

		Seite
	Bilderverzeichnis	8
1.	**Einführung**	9
1.1.	Vorbemerkung	9
1.2.	Technische Daten des Systems	9
2.	**Zusammenbau**	11
2.1.	Nötige Werkzeuge	11
2.2.	Druckschriften	11
2.3.	Bauteile	11
2.4.	Stückliste	13
2.5.	Vorgehen beim Zusammenbau	15
2.6.	Zusammenbau – Abschnitt 1	15
2.6.1.	Stückliste für Zusammenbau – Abschnitt 1	16
2.7.	Zusammenbau – Abschnitt 2	21
2.7.1.	Stückliste für Zusammenbau – Abschnitt 2	21
2.8.	Zusammenbau – Abschnitt 3	27
2.8.1.	Stückliste für Zusammenbau – Abschnitt 3	27
2.9.	Fädelstift	32
3.	**Erklärungen zur Arbeitsweise Ihres Mikrocomputers**	35
3.1.	Allgemeines	35
3.2.	System-Bus-Leitungen	36
3.3.	Rücksetztaste	36
3.4.	Taktgenerator	36
3.5.	$\phi 1$- und $\phi 2$-Taktsignale	36
3.6.	RESET-Signal	36
3.7.	READY-Signal	36
3.8.	OSC-Signal	37
3.9.	$\overline{\text{STSTB}}$-(Status Strobe) Signal	37
3.10.	Der Mikroprozessor SAB 8080A	37
3.11.	System-Steuer-Baustein SAB 8228	37
3.12.	Programm-Unterbrechungen („Interrupt")	38

Seite

3.13. Adreß-Puffer (wahlweise) 38
3.14. SIKIT-DK/8080-Speicherstruktur 38
3.14.1. ROM-Dekoder und Befehlsspeicher 39
3.14.2. RAM-Dekoder und Datenspeicher 40
3.15. Die Schaltung zur Festlegung der Übertragungsgeschwindigkeit 41
3.16. Serienschnittstelle mit SAB 8251 41
3.17. Übrige Ein-/Ausgabeleitungen 42

4. Endzusammenbau und Überprüfung 43
4.1. Allgemeines 43
4.2. Spezifische Anpassung des Computer-Systems 43
4.2.1. HOLD- und WAIT-Möglichkeiten 43
4.2.2. Freigabe der Busse 43
4.2.3. Festlegung der Datenübertragungsrate 44
4.2.4. Festlegung des Serienschnittstellenpegels 44
4.2.5. Adreß-Bus-Puffer 45
4.2.6. Anschluß der Ein-/Ausgabeleitungen 45
4.3. Einfügen der Mikrocomputer – IS 47
4.3.1. Einbauhinweise 47
4.3.2. Ausrichten der IS-Anschlüsse 48
4.3.3. Einsetzen der IS in die Computerplatine 48
4.3.4. Herausnehmen von IS aus ihren Fassungen 48
4.3.5. Einsetzen des Taktgenerators 48
4.3.6. Einsetzen der übrigen IS-Bausteine 49
4.4. Inbetriebnahme 52
4.5. Hinweise zur Fehlersuche 52

5. Das MONITOR-Programm 53
5.1. Allgemeines 53
5.2. Funktionsweise 53
5.2.1. Programmerstellung und -Test mit Hilfe des Monitors 53
5.2.2. Ein-/Ausgabeteil 54
5.3. Kommando-Formate 54
5.3.1. Anzeige des Speicherinhalts: Kommando „D" 54

Inhaltsverzeichnis

		Seite
5.3.2.	Programmstart: Kommando „G"	55
5.3.3.	Einschreiben eines Befehls in den RAM-Bereich: Kommando „I"	55
5.3.4.	Datentransfer im Speicher: Kommando „M"	56
5.3.5.	Änderung einer Speicherstelle: Kommando „S"	56
5.3.6.	Ausgabe und Änderung der Prozessor-Register: Kommando „X"	57
5.4.	**Ein-/Ausgabe**	57
5.4.1.	Die Eingaberoutine CI (Konsolen-Eingabe)	57
5.4.2.	Die Ausgaberoutine CO (Konsolen-Ausgabe)	58
5.5.	**Betriebshinweise**	58
5.5.1.	Aktivierung des Programms	58
5.5.2.	Stapelspeicherbereich	58
5.5.3.	Testmöglichkeit durch Aufsetzpunkt	58
5.5.4.	Unterbrechungstechnik („Interrupt")	59
5.6.	**Fehlermeldungen**	59
5.6.1.	Unzulässige Zeichen	59
5.6.2.	Adreß-Fehler	59
6.	**Ausdruck des Monitor-Primärprogrammes**	61
7.	**Gesamtschaltung des SIKIT-DK/8080-Computers**	89
8.	**Gesamtbestückungsplan**	97
	Anschriften unserer Geschäftsstellen	99

Bilderverzeichnis

Bild	Titel	Seite
1	Systemübersicht (Microcomputer-Bausteine)	12
2	Einlöten der Widerstände R1 bis R4 und R21 bis R24	17
3	Einlöten der Kondensatoren C1 bis C3 und C10 bis C16	18
4	Einlöten der Diode CR1, des Rücksetzschalters SW1, der Lötstützpunkte E1 bis E6, des Taktquarzes bei Position Y1 und der 16 pin-Fassung bei Position A8	19
5	Fertige Teilbestückung nach Abschnitt 1	20
6	Einlöten der Widerstände R5 bis R17, R19, R19', R20, R20', R25 und R26	23
7	Einlöten der Kondensatoren C4 bis C9 und C17 bis C22	24
8	Einlöten der Transistoren Q1, Q2, Q3 und der Diode CR2 sowie der 28 pin-Fassungen in die Positionen A7 und A9	25
9	Fertige Teilbestückung nach Abschnitt 2	26
10	Einlöten der Widerstände R27 bis R30 und der 24 pin-Fassungen in die Positionen A14 und A15	28
11	Einlöten des Kondensators C23 und der Kondensatoren C24 bis C35	29
12	Position des Spannungsreglers VR1	30
13	Einsetzen des Spannungsreglers VR1	30
14	Fertige Teilbestückung nach Abschnitt 3	31
15	Arbeiten mit Fädelstift	32
16	SIKIT-DK/8080 Blockschaltbild	35
17	Speicheraufteilung des SIKIT-DK/8080	38
18	ROM/PROM-Adreß-Format	39
19	RAM-Adreß-Format	40
20	Serienschnittstellen-Adreß-Format	41
21	Parallelschnittstellen-Adreß-Format	42
22	Einlöten bzw. Einsetzen der IS	50
23	Mikroprozessor	89
24	PROM-Bereich	91
25	RAM-Bereich	93
26	E/A-Baustein-Konfiguration	95
27	Gesamtbestückungsplan	97

1. Einführung

1.1. Vorbemerkung (Wichtig!)

SIKIT-DK/8080 ist ein vollständiges 8080-Mikrocomputer-System in Bausatzform, das sich in nur 6 Stunden zu einem voll funktionsfähigen System zusammenbauen läßt und sich als Starthilfe bei der Einarbeitung in die Mikrocomputer-Technik und als Prototyp für Geräteentwicklungen hervorragend eignet.

Durch die volle elektrische und mechanische Kompatibilität zum Entwicklungscomputer SME ist höchste Flexibilität, Ausbaufähigkeit und Testmöglichkeit gewährleistet.

Für den Datenverkehr mit dem Computer und für Testzwecke ist dem Bausatz ein bereits fertig programmiertes ROM mit einem System-Monitor-Programm beigefügt.

Zur Inbetriebnahme zusätzlich nötig sind lediglich Stromversorgung und ein Datenendgerät, wie z.B. Siemens BEA 3913; die hierfür nötige Anpassungselektronik ist bereits im Bausatz enthalten.

Die folgende Bauanleitung ist vor Beginn der Zusammenbau-Arbeiten genau durchzulesen und dann beim Aufbau des Systems in der vorgegebenen Reihenfolge anzuwenden, um den erfolgreichen Aufbau des Systems zu garantieren.

1.2. Technische Daten des Systems

Der serienmäßig gelieferte SIKIT-DK/8080 umfaßt folgende Systemkomponenten:

– Mikroprozessor-(Zentraleinheit):	1 × SAB 8080A
– Programmspeicher:	2 × SAB 8708 EPROM's, von denen eines mit dem DK-Monitor programmiert geliefert wird (≙ 2 KByte-Programmspeicher)
– Lese/Schreibspeicher:	2 × SAB 8111-1 statische RAM's (≙ $^1/_4$ KByte-Variablen-Speicher)
– Ein/Ausgabeeinheiten:	1 × SAB 8255 Programmierbare Peripherie-Schnittstelle (PPI) mit 24 Ein/Ausgabeleitungen 1 × SAB 8251 Programmierbare Serien-Schnittstelle (PSS) für Anschluß von Datensichtgeräten oder Konsole BEA 3913
– Übertragungsrate:	75, 110, 300, 600, 1 200, 2400, 4800 Baud
– Programm-Unterbrechungsbehandlung (Interrupt):	1 Unterbrechungsebene (RST 7)

Ohne jegliche Veränderung der Leiterplatine ist durch Einfügen zusätzlicher Mikrocomputer-Bausteine folgende Erweiterung des SIKIT-DK/8080 möglich:

– Programmspeicher	erweiterbar bis 4 KByte durch 2 zusätzliche Bausteine SAB 8708
– Lese/Schreibspeicher	erweiterbar bis 1 KByte durch 6 zusätzliche Bausteine SAB 8111-1
– Ein/Ausgabe-Leitungszahl	erweiterbar auf 48 parallele Ein/Ausgabe-Leitungen durch 1 zusätzlichen Baustein SAB 8255

2. Zusammenbau

2.1. Nötige Werkzeuge

Bevor Sie beginnen, sollten Sie folgende Werkzeuge bereit legen:
- 1 kleine Flachzange
- 1 kleiner Schraubenzieher (ca. 2 mm Klingenbreite)
- 1 Schraubenzieher (ca. 6 mm Klingenbreite)
- 1 kleiner Seitenschneider
- 1 Lötkolben 25 W
- Radio-Lötzinn mit Flußmittel-Seele, ca. 60:40 Zinngehalt (bitte kein anderes Lötzinn verwenden!!)
- 1 Multimeter (Volt/mA/Ω)
- 1 Zweistrahloszillograph (nicht unbedingt notwendig)

2.2. Druckschriften

Außer der Bauanleitung, die Sie vor sich haben, enthält Ihr SIKIT-DK/8080-Bausatz noch folgende Druckschriften:
- ASSEMBLER-Programmiersprache, System SAB 8080
- PL/M-Programmiersprache, System SAB 8080
- Mikroprozessor-Bausteine Datenbuch, System SAB 8080

2.3. Bauteile

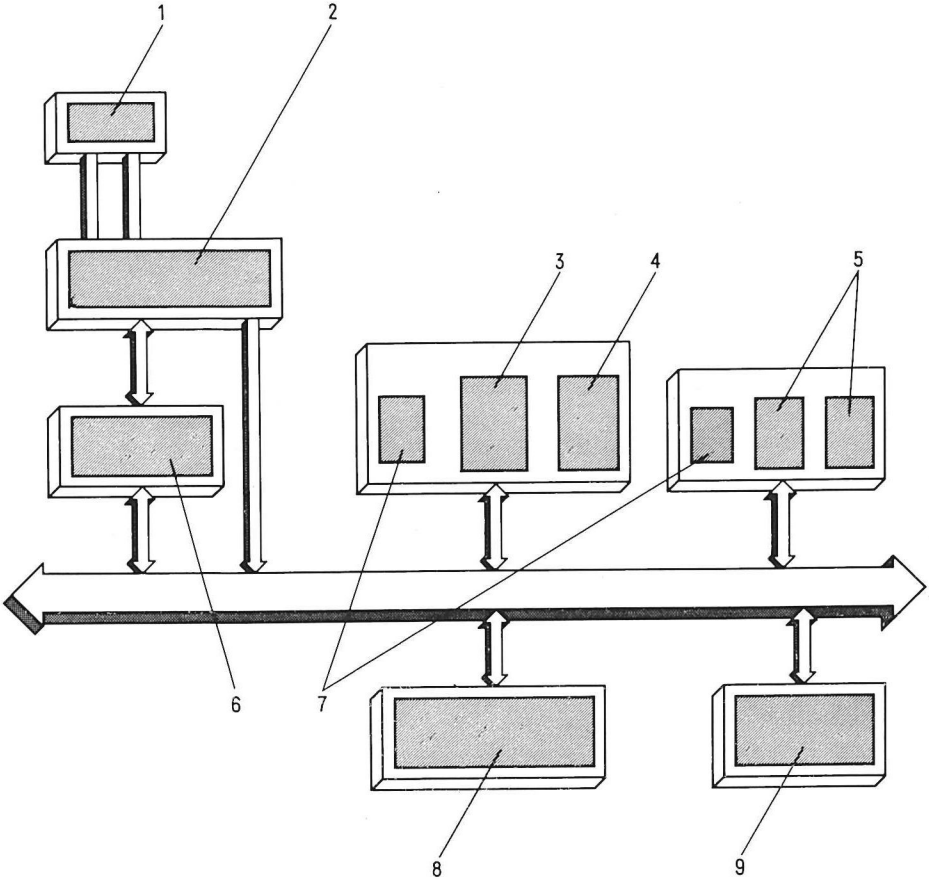
Neben den Druckschriften finden Sie folgende elektronischen Bauelemente vor:
- 1 gedruckte Leiterplatte, SME-kompatibel
- 1 Beutel mit verschiedenen IS-Bausteinen, Transistoren und passiven Bauelementen
- 1 Systemübersicht mit den eingesteckten Mikrocomputer-Bausteinen

Achtung!

Bei den Mikrocomputer-Bausteinen handelt es sich überwiegend um MOS-Bausteine, die trotz ihrer integrierten Schutzstrukturen mit Vorsicht gehandhabt werden sollten.

Bitte nehmen Sie die Bausteine deshalb erst dann aus dem leitenden Schaumstoff heraus, wenn es durch einen Hinweis in der Bauanleitung nötig ist!

Bild 1
Systemübersicht (Mikrocomputer-Bausteine)



1 = SAB 8224 (Taktgeber und Treiber)
2 = SAB 8080A (1-Chip 8-Bit-n-Kanal-Mikroprozessor)
3 = SAB 8708 (Löschbarer und neu programmierbarer Festspeicher EPROM)
4 = SAB 8308 mit SDK/8080-Monitor programmiert
5 = SAB 8111-1 (Statischer Schreib-Lese-Speicher)
6 = SAB 8228 (System-Steuer- und Bus-Treiber)
7 = SAB 8205 (Schneller 1-aus-8-Binär-Dekoder)
8 = SAB 8255 (Programmierbare, periphere Schnittstelle)
9 = SAB 8251 (Programmierbare Serienschnittstelle)

2.4. Stückliste

Führen Sie zunächst eine kleine Bestandsaufnahme in ihrem Bausatz durch.

Bitte nehmen Sie die genannten Bauteile vorerst auf keinen Fall aus dem Kit heraus!

Stückliste des Bausatzes SIKIT-DK/8080

Stück	Teil	Bezeichnung	In Bild 1 Nr.
Mikrocomputer-IS			
1	SAB 8080A	1-Chip 8-Bit-n-Kanal-Mikroprozessor	2
2	SAB 8111-1	Statische Schreib-Lese-Speicher	5
2	SAB 8205	Schnelle 1-aus-8-Binär-Dekoder	7
1	SAB 8224	Taktgeber und Treiber	1
1	SAB 8228	System-Steuer- und Bus-Treiber	6
1	SAB 8251	Programmierbare Serienschnittstelle	9
1	SAB 8255	Programmierbare periphere Schnittstelle	8
1	SAB 8708	Löschbarer und neu programmierbarer Festspeicher (EPROM)	3
1	SAB 8308	mit SDK-8080-Monitor programmiert	4
übrige Bauteile			
2	IS FLJ 411		
1	IS FLH 481		
1	IS TDB 7905T Spannungsregler		
1	IS 74S161		
2	IS-Fassungen 40 pol.		
2	IS-Fassungen 24 pol.		
2	IS-Fassungen 28 pol.		
1	IS-Fassung 16 pol.		
2	Steckverbinder 25 pol.		
2	Stecker 25 pol.		
1	SIEMENS-Fädelstift		
1	Druckschalter (Nr. C 42315-A60-A3)		
1	Quarz, 18.432 MHz		

Stück	Teil			
1	Widerstand,	10 kΩ,	1/4 W	(braun-schwarz-orange)
16	Widerstände,	1 kΩ,	1/4 W	(braun-schwarz-rot)
1	Widerstand,	2,7 kΩ,	1/4 W	(rot-violett-rot)
2	Widerstände,	47 Ω,	1/4 W	(gelb-violett-schwarz)
1	Widerstand,	390 Ω,	1/4 W	(orange-weiß-braun)
4	Widerstände,	220 Ω,	0,8 W	(rot-rot-schwarz-schwarz)
1	Widerstand,	4,7 kΩ,	1/4 W	(gelb-violett-schwarz-braun)
1	Widerstand,	560 kΩ,	1/4 W	(grün-blau-gelb)
2	Widerstände,	5,1 kΩ,	1/4 W	(grün-braun-rot)
1	Widerstand,	150 Ω,	1/4 W	(braun-grün-braun)
1	Widerstand,	510 Ω,	1/4 W	(grün-braun-braun)
1	Widerstand,	120 Ω,	1/4 W	(braun-rot-braun)
2	Dioden BAY 61			
2	Transistoren	BC 107		
1	Transistoren	BC 177		
2	Tantalelko,	1 µF, 40 V		
3	Tantalelko,	22 µF, 16 V		
1	Keramischer-Scheiben-Kondensator,	10 pF		
13	Keramische-Scheiben-Kondensatoren,	0,1 µF, 63 V		
16	Keramische-Scheiben-Kondensatoren,	10 nF, 32 V		
6	Steckstifte, 1,3 mm			
6	Kabelschuhe			
15	Abstandsscheiben (Kunststoff) für M3			
5	Gummifüße			
4	Schrauben	M3 × 8		
5	Schrauben	M3 × 18		
10	Muttern	M3		
2	Clips			
5	Abstandsrollen für M3, 10 mm			

Nachdem Sie sich vergewissert haben, daß alle Teile vorhanden sind, können Sie nun die Platine aufbauen.

2.5. Vorgehen beim Zusammenbau

Um Fehler soweit wie möglich auszuschalten, haben wir den Zusammenbau in drei Abschnitte unterteilt. Für jeden Abschnitt sind Bestückungspläne und eine entsprechende Bauanleitung vorhanden.
Bitte beachten Sie, daß die Bohrungen in der Leiterplatine auf keinen Fall nachgebohrt werden dürfen, da sonst die Durchkontaktierungen zerstört werden. Sollte dies durch ungünstige, mechanische Toleranzen doch einmal unumgänglich werden, muß dafür gesorgt sein, daß die Ober- und Unterseiten der Platine durch ein beidseitig verlötetes Drähtchen miteinander verbunden sind.

2.6. Zusammenbau – Abschnitt 1 (Bilder 2 bis 5)

Verfahren Sie nun bitte wie folgt:
- Legen Sie die Leiterplatte mit dem Firmenaufdruck („Siemens SIKIT-DK/8080") nach oben auf Ihren Arbeitstisch (Bestückungsseite).
- Bringen Sie zunächst die 5 Gummifüße an, einen in der Mitte und einen in jeder Ecke der Leiterplatte. (Schraube, Gummifuß und Isolier-Beilagscheibe von unten einstecken. Die Schraube von oben mit einer Mutter befestigen.)
- Löten Sie die Widerstände R1 bis R4 und R21 bis R24 entsprechend Bild 2 ein.
- Löten Sie die Kondensatoren C1 bis C3 und C10 bis C16 entsprechend Bild 3 ein.
  Beachten Sie die Polarität (+) der Elektrolyt-Kondensatoren C1, C15 und C16. („+"-Zeichen ist auf der Platine vorgedruckt.)
- Löten Sie die Diode CR1 ein (Bild 4, Polarität beachten!).
- Löten Sie den Rücksetzschalter SW1 ein (Bild 4).
- Löten Sie die Lötstützpunkte (Steckstifte) E1 bis E6 ein (Bild 4).
- Bringen Sie den Taktquarz auf der Platine bei Y1 an (Bild 4); die Anschlußdrähte müssen hierzu mit Hilfe einer kleinen Flachzange rechtwinkelig abgebogen werden.
- Löten Sie die 16 pin-Fassung bei Position A8 ein (Bild 4).

Sie haben nun den ersten Abschnitt fertig aufgebaut. Vergleichen Sie ihr Ergebnis mit Bild 5.

2.6.1. Stückliste für Zusammenbau – Abschnitt 1

Bauteil-bezeichnung	Bauteil	Wert
R1	Widerstände	120 Ω, 1/4 W (braun-rot-braun)
R2		1 kΩ, 1/4 W (braun-schwarz-rot)
R3		1 kΩ, 1/4 W (braun-schwarz-rot)
R4		560 kΩ, 1/4 W (grün-blau-gelb)
R21		1 kΩ, 1/4 W (braun-schwarz-rot)
R22		5,1 kΩ, 1/4 W (grün-braun-rot)
R23		1 kΩ, 1/4 W (braun-schwarz-rot)
R24		1 kΩ, 1/4 W (braun-schwarz-rot)
C1	Tantal-Elko	1 µF 40 V
C2	Keramische Scheiben-Kondensatoren	10 nF
C3		10 nF
C10		10 pF
C11		10 nF
C12		10 nF
C13		10 nF
C14		10 nF
C15	Tantal-Elko's	22 µF 16 V
C16		22 µF 16 V
CR1	Diode BAY 61	
SW1	Rücksetzschalter	
E1	Lötstützpunkte	
E2		
E3		
E4		
E5		
E6		
Y1	Quarz	
A8	16 pin-Sockel	

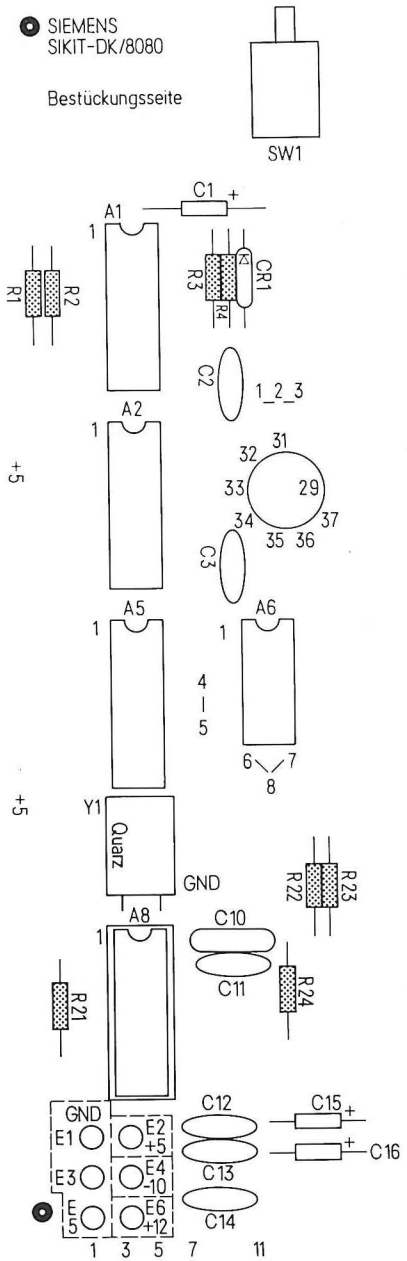


Bild 2
Einlöten der Widerstände R1 bis R4 und R21 bis R24

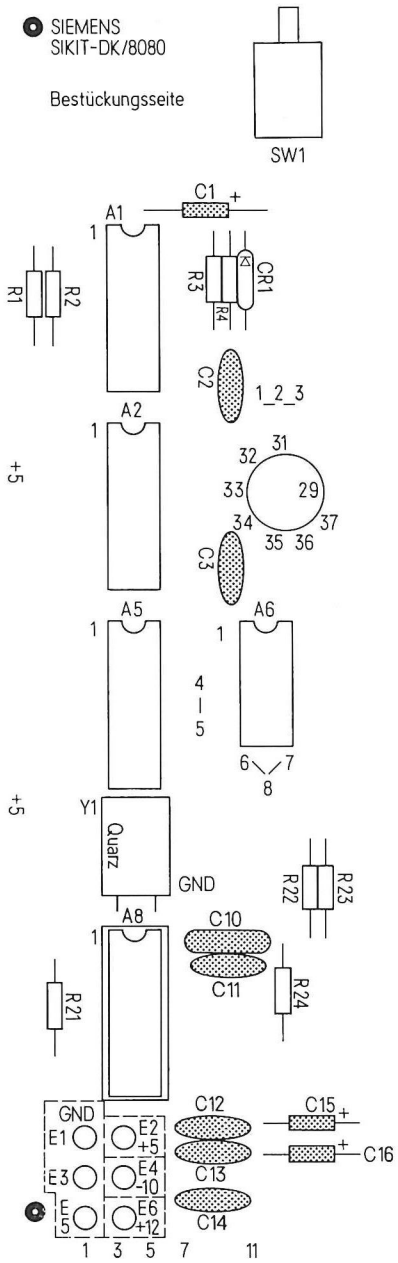


Bild 3
Einlöten der Kondensatoren C1 bis C3 und C10 bis C16

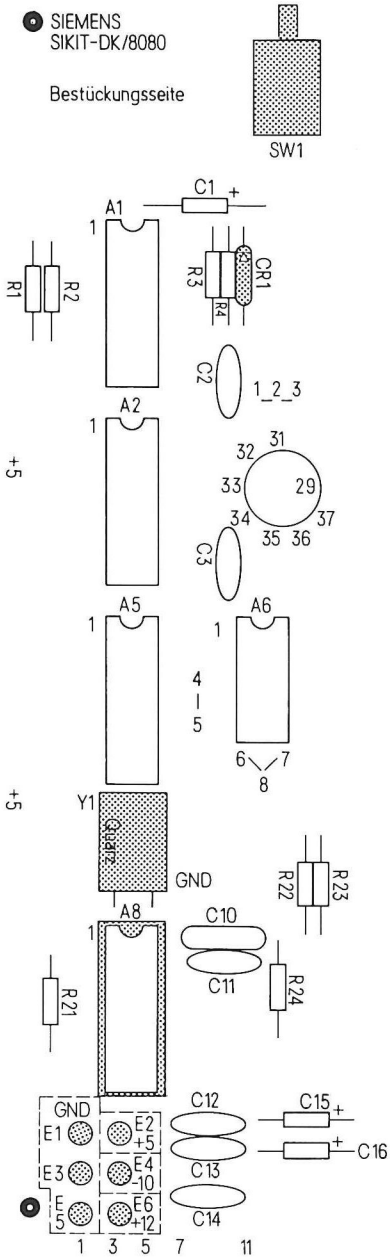


Bild 4
Einlöten der Diode CR1, des Rücksetzschalters SW1, der Lötstützpunkte E1 bis E6, des Taktquarzes bei Position Y1 und der 16 pin-Fassung bei Position A8

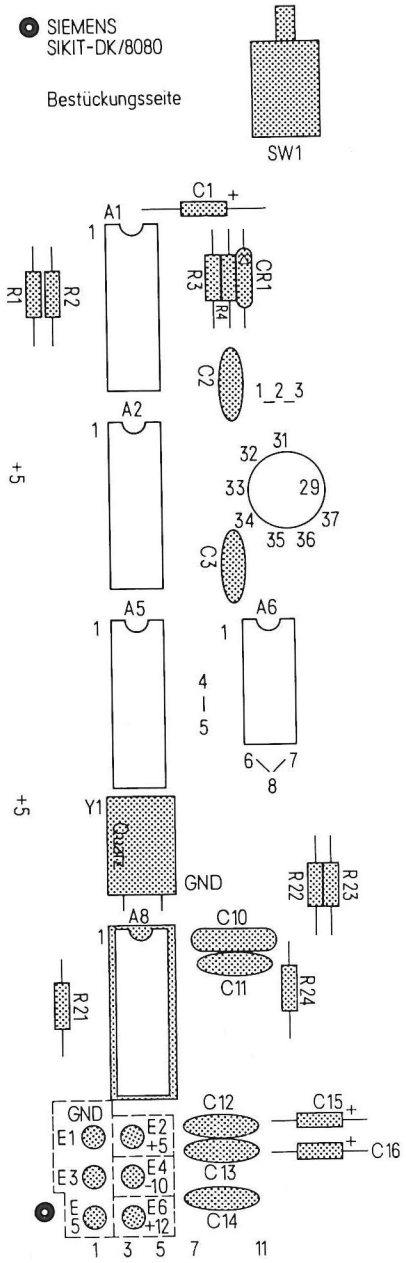


Bild 5
Fertige Teilbestückung nach Abschnitt 1

2.7. Zusammenbau – Abschnitt 2 (Bilder 6 bis 9)

- Löten Sie die Widerstände R5 bis R17, R19, R19′, R20, R20′, R25 und R26 entsprechend Bild 6 ein.
- Löten Sie die Kondensatoren C4 bis C9 und C17 bis C22 entsprechend Bild 7 ein.
  Beachten Sie auch hier, daß die Elektrolyt-Kondensatoren C21 und C22 so eingesetzt sind, daß ihre „+"-Enden in der gleichen Position sind wie auf der Leiterplatte vorgedruckt.
- Löten Sie die Transistoren Q1, Q2 und Q3 entsprechend Bild 8 ein.
  Beachten Sie dabei, daß sich die Gehäusenasen in der dem Platinenaufdruck entsprechenden Stelle befinden.
- Löten Sie die Diode CR2 ein und achten Sie auf die Polarität (Bild 8).
- Löten Sie die 28 pin-Fassungen in die Positionen A7 und A9 und die 40 pin Fassungen in die Positionen A3 und A10 ein (Bild 8).
- Setzen Sie einen 25 pol. Steckverbinder („Weibchen") in die Position J1 und befestigen Sie ihn mit 2 Schrauben, Isolierscheiben und Muttern. Löten Sie ihn jetzt von unten an.

Damit ist Abschnitt 2 fertig. Überprüfen Sie das Ergebnis anhand von Bild 9.

2.7.1. Stückliste für Zusammenbau – Abschnitt 2

Bauteil-bezeichnung	Bauteil	Wert	
R5	Widerstände	5,1 kΩ, 1/4 W	(grün-braun-rot)
R6		4,7 kΩ, 1/4 W	(gelb-violett-schwarz-braun)
R7		1 kΩ, 1/4 W	(braun-schwarz-rot)
R8		10 kΩ, 1/4 W	(braun-schwarz-orange)
R9		2,7 kΩ, 1/4 W	(rot-violett-rot)
R10		1 kΩ, 1/4 W	(braun-schwarz-rot)
R11		47 Ω, 1/4 W	(gelb-violett-schwarz)
R12		150 Ω, 1/4 W	(braun-grün-braun)
R13		siehe Bemerkung!	(nächste Seite)
R14		47 Ω, 1/4 W	(gelb-violett-schwarz)
R15		1 kΩ, 1/4 W	(braun-schwarz-rot)
R16		1 kΩ, 1/4 W	(braun-schwarz-rot)
R17		1 kΩ, 1/4 W	(braun-schwarz-rot)
R19		220 Ω, 0,8 W	(rot-rot-schwarz)
R19′		220 Ω, 0,8 W	(rot-rot-schwarz)
R20		220 Ω, 0,8 W	(rot-rot-schwarz)
R20′		220 Ω, 0,8 W	(rot-rot-schwarz)
R25		1 k, 1/4 W	(braun-schwarz-rot)
R26		1 k, 1/4 W	(braun-schwarz-rot)

Bauteil-bezeichnung	Bauteil	Wert
C4	Keramische Scheiben-Kondensatoren	10 nF
C5		0,1 µF
C6		10 nF
C7		10 nF
C8		10 nF
C9		10 nF
C17		10 nF
C18		10 nF
C19		10 nF
C20		10 nF
C21	Tantal-Elko's	1 µF 40 V
C22		22 µF 16 V
Q1	Transistoren	BC 107
Q2		BC 107
Q3		BC 177
CR2	Diode	BAY 61
A3	IS-Fassungen	40 pin.
A7		28 pin.
A9		28 pin.
A10		40 pin.
I1	Steckverbinder	25 pol.

Bemerkung: Mit R13 läßt sich der Wert der negativen Versorgungsspannung einstellen. Sein Wert ist 390 Ω (1/4 W, Farbkennung orange/weiß/braun) für −10 V bzw. 510 Ω (1/4 W, grün/braun/braun) für 12...15 V.

Bild 6
Einlöten der Widerstände R5 bis R17, R19, R19′, R20, R20′, R25 und R26

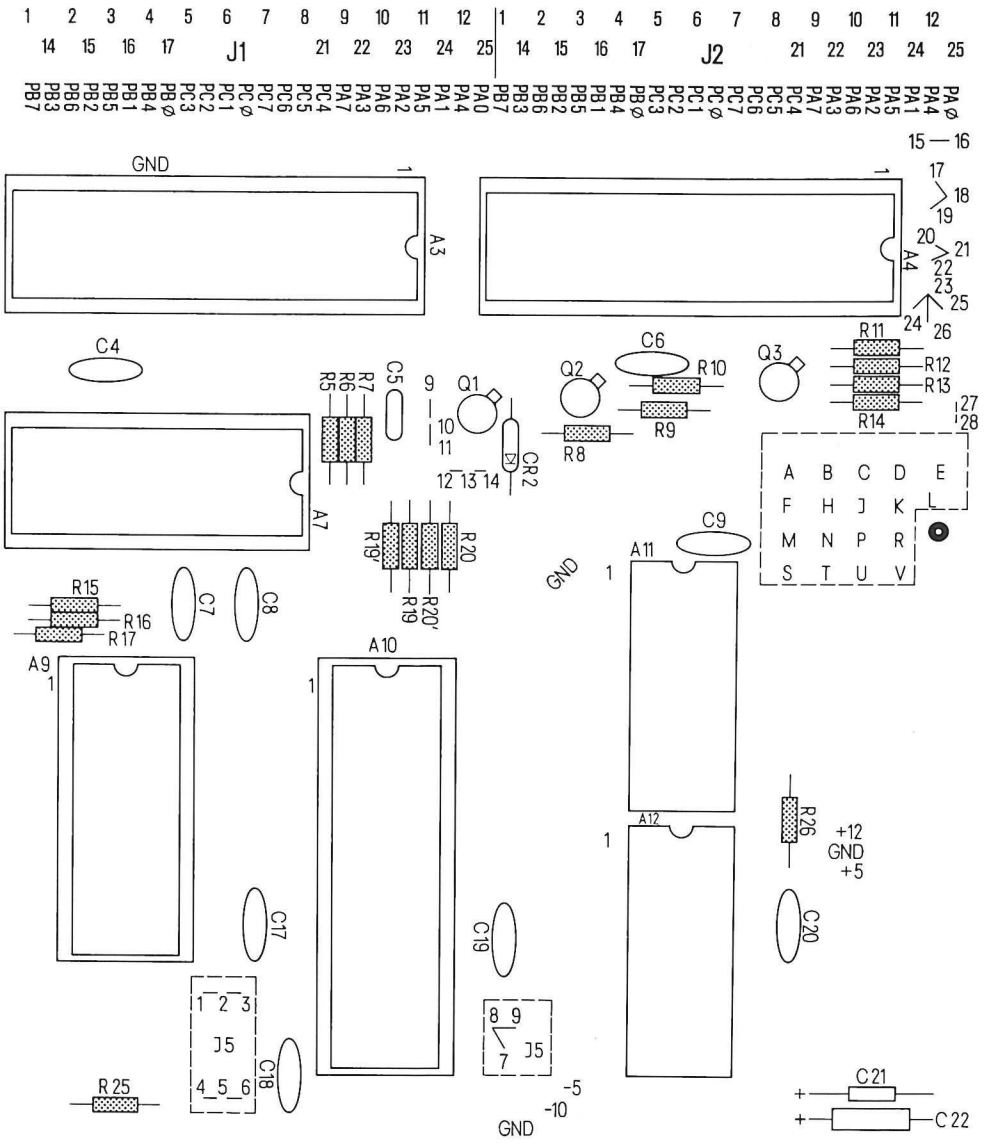


Bild 7
Einlöten der Kondensatoren C4 bis C9 und C17 bis C22

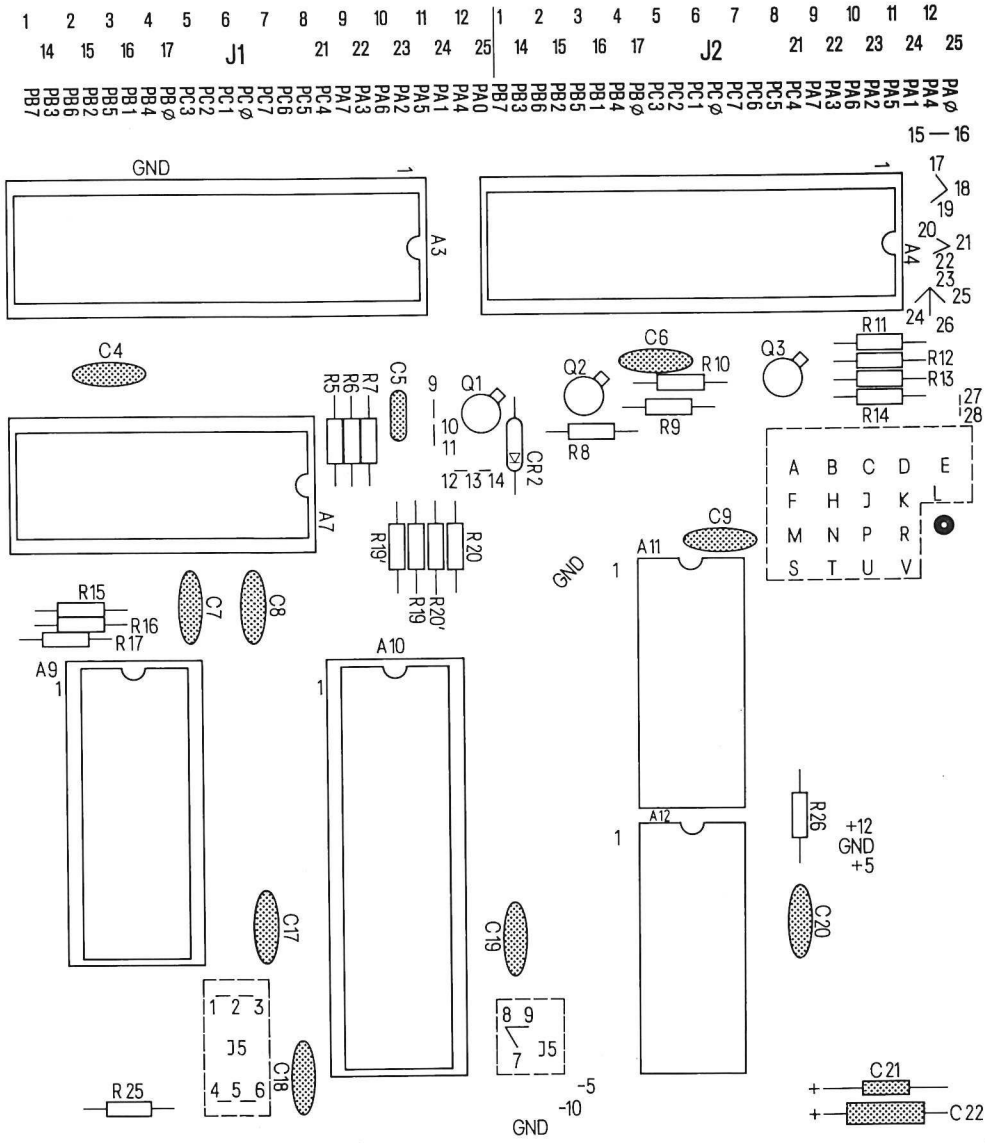


Bild 8
Einlöten der Transistoren Q1, Q2, Q3, der Diode CR2 sowie der 28 pin-Fassungen in die Positionen A7 und A9 und der 40 pin-Fassungen in die Positionen A3 und A10.

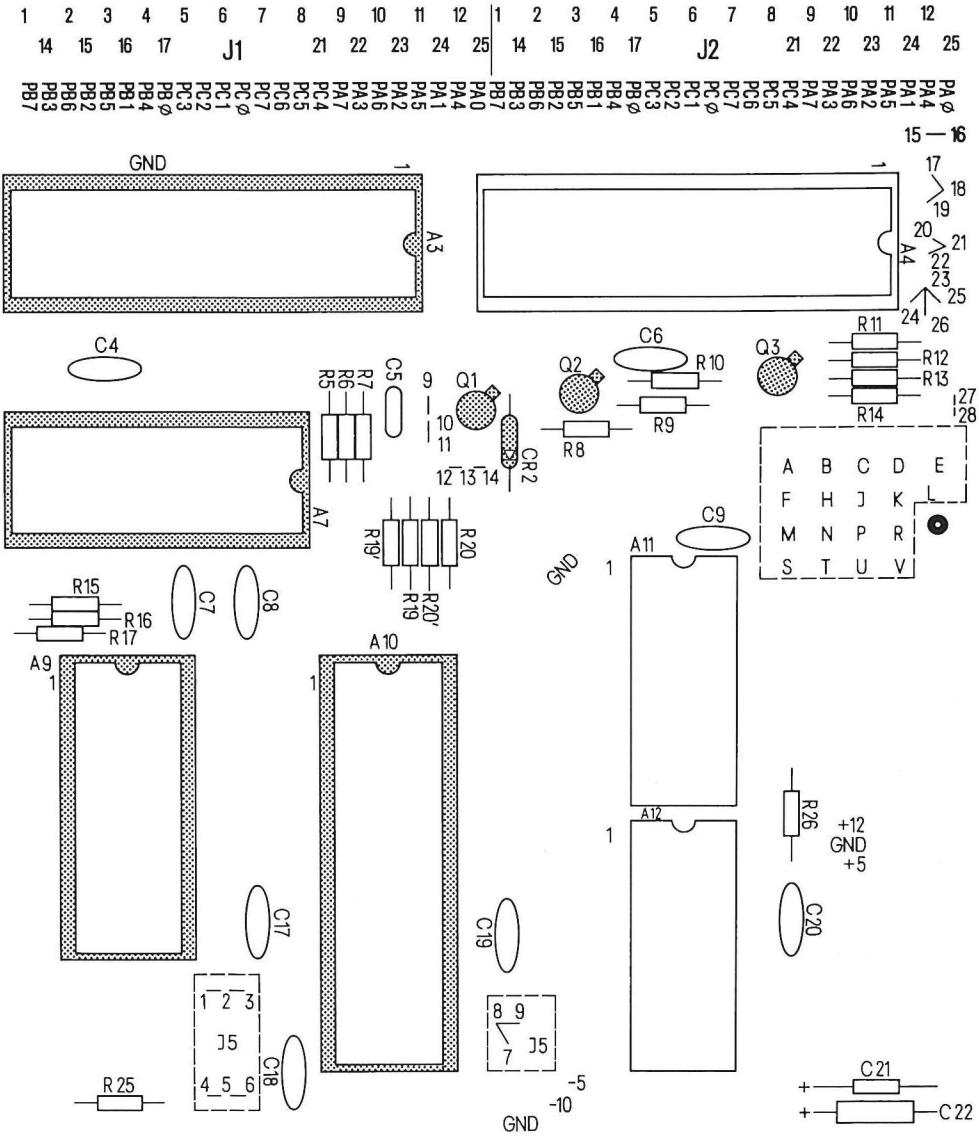
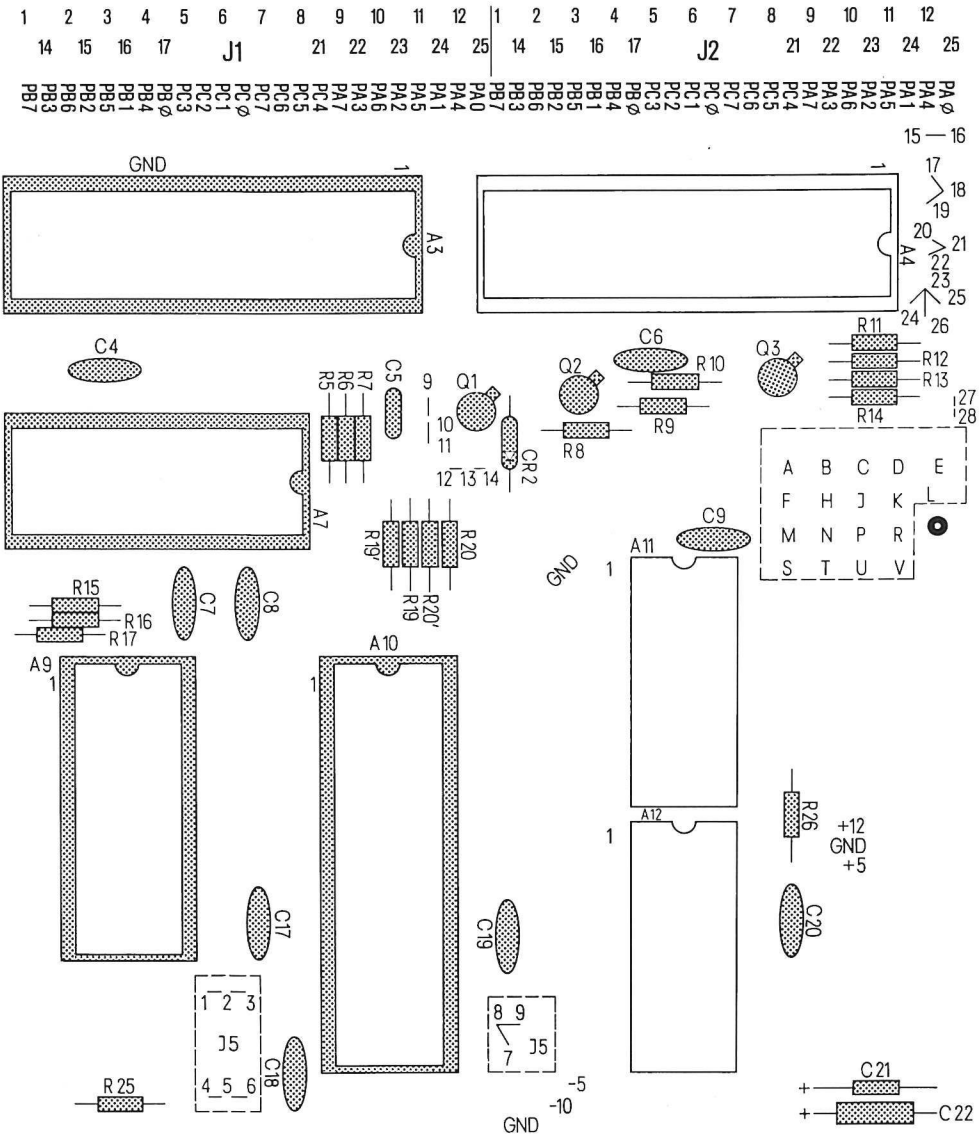


Bild 9
Fertige Teilbestückung nach Abschnitt 2



2.8. Zusammenbau – Abschnitt 3 (Bilder 10 bis 14)

- Löten Sie die Widerstände R27 bis R30 und die 24 pin-Fassungen in die Positionen A14 und A15 ein (Bild 10).
- Löten Sie den Kondensator C23 und die Kondensatoren C24 bis C35 ein (Bild 11).
- Setzen Sie den Spannungsregler VR1 in die vorgesehene Position in der Platine ein (Bild 12). Die Anschlußdrähte müssen dazu mit einer Flachzange vorsichtig rechtwinklig abgebogen werden (Bild 13).
- Befestigen Sie nun den Spannungsregler entsprechend Bild 13 und löten Sie ihn von der Unterseite her ein.
- Stecken Sie einen 25 pin-Steckverbinder („Weibchen") in Position J3, befestigen Sie ihn mit 2 Schrauben, 2 Isolierscheiben und 2 Muttern und löten Sie ihn von unten ein.

Damit ist Abschnitt 3 fertig. Überprüfen Sie das Ergebnis anhand von Bild 14.

2.8.1. Stückliste für Zusammenbau – Abschnitt 3

Bauteil-bezeichnung	Bauteil	Wert
R27 R28 R29 R30	Widerstände	1 kΩ, 1/4 W (braun-schwarz-rot)
C23	Keramischer Scheiben-Kondensator	10 nF
C24 C25 C26 C27 C28 C29 C30 C31 C32 C33 C34 C35	Kondensatoren	0,1 µF 50 V
A14 A15	IS-Fassungen	24 pin
J3	Steckverbinder („Weibchen")	25 pol.
VR1	Spannungsregler (IS)	TDB 7905T

Bild 10
Einlöten der Widerstände R27 bis R30 und der 24 pin-Fassungen in die Positionen A14 und A15

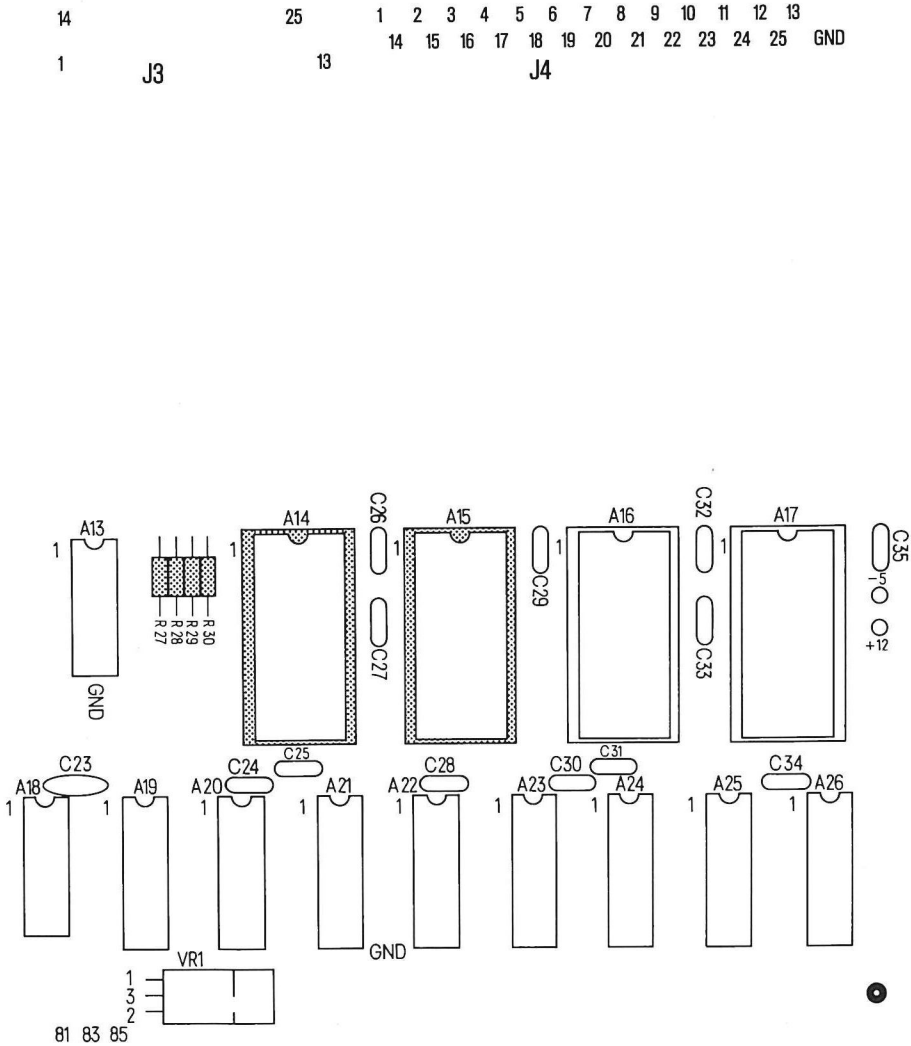


Bild 11
Einlöten des Kondensators C23 und der Kondensatoren C24 bis C35

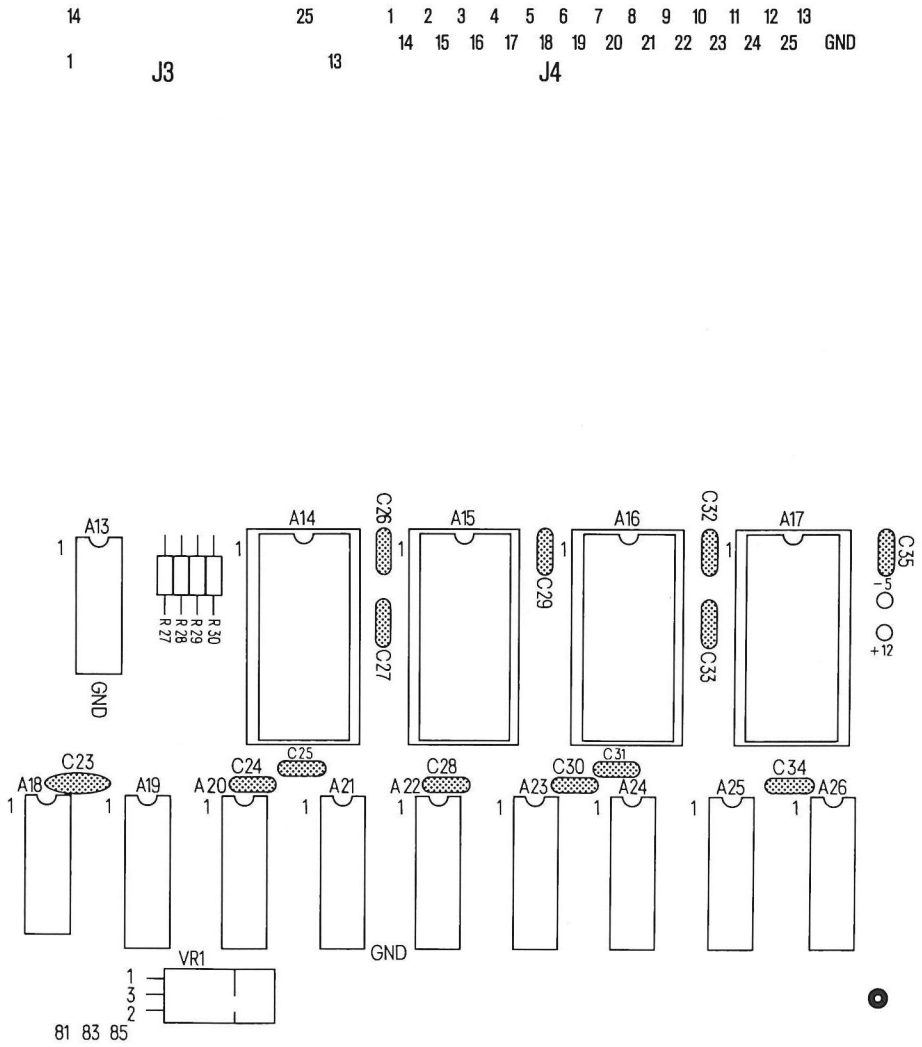


Bild 12
Position des Spannungsreglers VR1

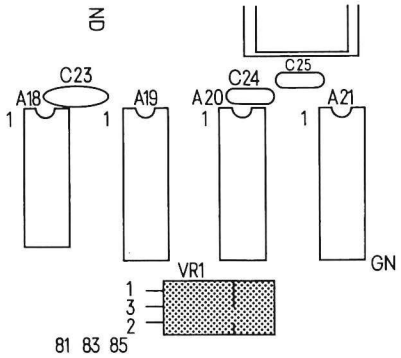


Bild 13
Einsetzen des Spannungsreglers VR1

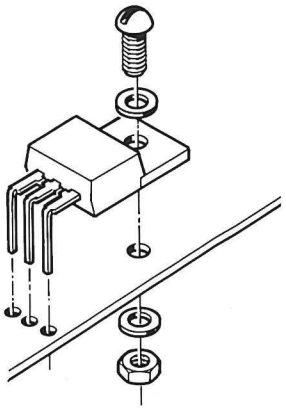
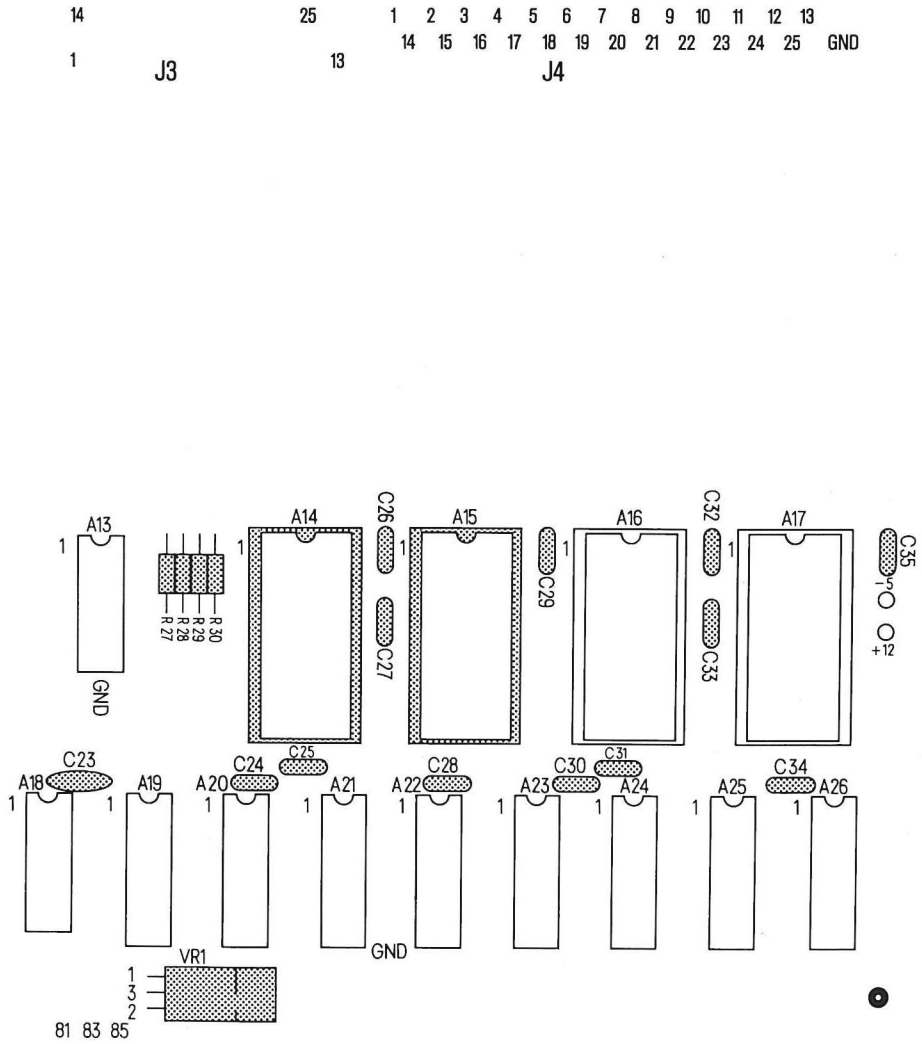


Bild 14
Fertige Teilbestückung nach Abschnitt 3



2.9. Fädelstift

Der Fädelstift dient zur Herstellung von Verbindungen auf der freizuverdrahtenden, anwenderspezifischen Platinenfläche des SIKIT-DK/8080.

Der Vorteil der Fädeltechnik besteht in der schnellen Ausführung von Versuchsschaltungen ohne Verwendung geätzter Leiterbahnen.

Der Fädelstift trägt eine Rolle mit lackisoliertem, lötbarem Cu-Draht. Die Isolationsschicht schmilzt beim Lötvorgang. Von der Rolle läuft der Draht durch den Fädelstift, tritt an der Stiftspitze aus und wird an einen gewünschten Anschluß geführt. Bei komplexeren Verbindungen (z.B. sollen mehrere IS eingefügt werden) empfiehlt sich der Einsatz von Kunststoff-Fädelkämmen (nicht im Bausatz enthalten), an denen die einzelnen Leitungen festgelegt werden und so eine übersichtliche Verdrahtung garantiert ist.

Beim Fädelvorgang bildet z.B. ein Stift der Anschlußleiste den Ausgangspunkt, an dem der Draht zweckmäßigerweise angelötet wird. Von hier aus kann zu beliebigen Anschlußpunkten hin gefädelt werden, wobei der Draht lediglich mit ein paar Windungen um den erreichten Stift des jeweiligen Bauteils gewickelt, anschließend weitergeführt oder abgeschnitten wird.

Die umwickelten Anschlüsse werden erst nach komplett gefädelter Platte fest verlötet.

Technische Daten

Drahtart	doppelt isolierter lötfähiger Lackdraht Cu-LL rt V1 Typ 28F
Drahtdurchmesser	0,2 mm
Dicke der Isolationsschicht	13 µ
Spannungsfestigkeit	650 V (50 V/µ Isolationsdicke)
Drahtwiderstand	1 Ω/1,8 m
Länge der Drahtrolle	50 m
Thermische Dauerbelastung	5 A/mm², Gleichstrom
Störsicherheit	0,4 V/8 cm Drahtparallel-Führung, max. zul. Frequenz 3,2 MHz

Bestellbezeichnungen

Produktbezeichnungen	Bestellnummer
Fädelstift	C 26407-A36-A1
Drahtrolle für Fädelstift	C 26407-A36-C2
Stromschiene (nur 30-Stück-Packung)	C 26117-A268-C2
Fädelkamm, Kerbe rund (nur 30-Stück-Packung)	C 26117-A268-C4
Fädelkamm, Kerbe schwalbenschwanzförmig (nur 30-Stück-Packung)	C 26117-A270-C4

◁ **Bild 15**
Arbeiten mit Fädelstift

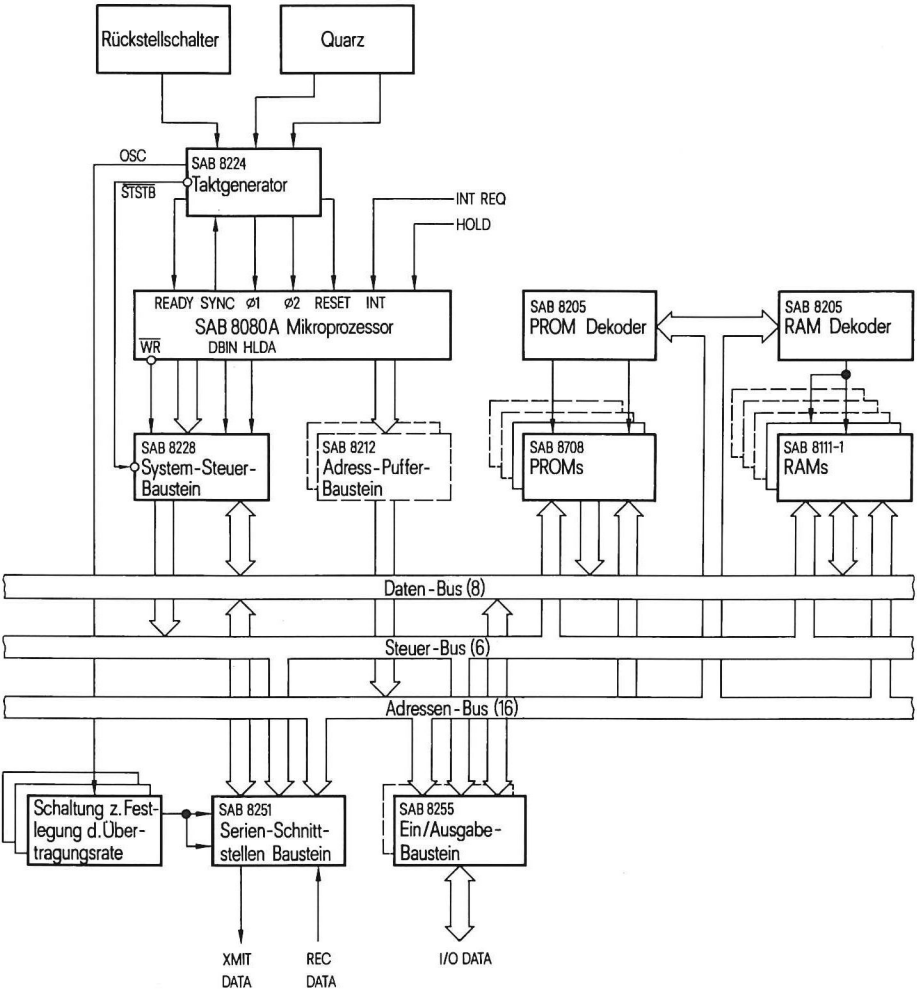
blank page

3. Erklärungen zur Arbeitsweise Ihres Mikrocomputers

3.1. Allgemeines

Vor Inbetriebnahme möchten wir Ihnen noch die Funktionsweise des nun fertig zusammengebauten Computers erläutern.

Bild 16
SIKIT-DK/8080 Blockschaltbild



Dieses funktionelle Block-Schaltbild des SIKIT-DK/8080, wurde zur Einführung absichtlich so einfach wie möglich gehalten; Sie werden feststellen, daß in diesem Bild nur die wichtigsten Signale des Systems gezeigt sind.

3.2. System-Bus-Leitungen

Die wesentliche Funktion des Computers spielt sich über 3 verschiedene Sammelschienen („Busse") ab, den Daten-Bus (8-Bit-parallel), den Adreß-Bus (16-Bit-Parallel) und den Steuersignal-Bus (6-Bit-parallel).

Die Sammelschienen können übrigens auf Anforderung durch das Anwendersystem gesperrt werden, soweit entsprechend Kapitel 4 ein derartiger Zugriff auf der Platine vorgesehen wird.

Hier soll noch darauf hingewiesen werden, daß das Datenbuch „Mikroprozessor-Bausteine" für sämtliche Einzelprobleme detaillierte Hinweise gibt.

3.3. Rücksetztaste

Die Rücksetztaste gibt Ihnen die Möglichkeit, das System zu jeder Zeit in einen definierten Anfangszustand zu bringen. Wenn die Taste gedrückt wird, sendet der Takt-Generator dem ganzen System ein RESET-Signal. Sie ist grundsätzlich nach Einschalten der Speisespannungen zu betätigen.

3.4. Taktgenerator

Der SAB 8224-Taktgenerator liefert das Zeitnormal des Systems, erzeugt das 12 V-Taktsignal für den SAB 8080-Mikroprozessor, synchronisiert das READY-Signal und erzeugt das Einschalt- und Rückstell-RESET-Signal.

3.5. Φ1- und Φ2-Taktsignale

Φ1 und Φ2 sind die 2,048 MHz-Taktsignale für den Mikroprozessor. Sie werden vom Oszillator durch Teilung der Quarzfrequenz durch 9 abgeleitet.

3.6. RESET-Signal

Bei jedem Einschalten der Versorgungsspannungen wird zunächst ein RESET-Signal an das System weitergegeben. Durch RESET wird der Befehlszähler des Mikroprozessors auf Null gesetzt, der RAM-Dekoder gesperrt und die Serienschnittstelle zurückgesetzt. Das gleiche Signal RESET wird dem System auch durch Betätigen der Rücksetztaste zur Verfügung gestellt (über Anschluß V auch extern zugänglich).

3.7. READY-Signal

Über READY ist eine externe Synchronisation des Mikroprozessors durch den Eingang RDYIN (Anschluß P) möglich.

3.8. OSC-Signal

OSC ist ein 18,432 MHz-Signal, das der Übertragungsraten-Schaltung eingespeist wird. Dadurch kann der Mikroprozessor auf maximaler Geschwindigkeit arbeiten und verfügt über eine stabile Basis für die Übertragungsraten-Schaltung.

3.9. $\overline{\text{STSTB}}$-(Status Strobe-)Signal

Am Anfang jedes Maschinenzyklus gibt der Mikroprozessor eine Status-Information auf seinen Daten-Bus aus. $\overline{\text{STSTB}}$ veranlaßt den SAB 8228-System-Steuer-Baustein zur Speicherung der Status-Information in seinem Status-Speicher. Zur Verwendung in externen Schaltungen ist $\overline{\text{STSTB}}$ dem Anwendersystem als STATUS STROBE-Signal an Anschluß J zugänglich gemacht.

3.10. Der Mikroprozessor SAB 8080A

Der Mikroprozessor benötigt einen Zwei-Phasen-Takt ($\Phi 1$ und $\Phi 2$) zwischen 0,5 MHz und 2,048 MHz. (Er wird vom Takt-Generator erzeugt.)

Auf dem 8-Bit-parallelen Daten-Bus liegen zweierlei Signale im Multiplexverfahren:

- die eigentlichen Ein-/Ausgabedaten, die der Mikroprozessor von Speichern und/oder Peripherie empfängt bzw. an sie sendet
- die Status-Informationen, die über den momentanen, internen Prozessorzustand Auskunft geben.

Da den übrigen System-Bausteinen die beiden Signaltypen in definierter Form getrennt verfügbar gemacht werden müssen, werden sie vom System-Steuer-Baustein SAB 8228 entsprechend aufbereitet.

Über den 16-Bit-parallelen Adreß-Bus werden den Speichern und Ein-/Ausgabe-Bausteinen die nötigen Adressier- und Auswahlinformationen angeboten. Zur Erhöhung der Belastbarkeit der Adreßleitungen, sind zwischen dem Mikroprozessor und den übrigen Bausteinen, Pufferschaltkreise einzufügen. An dieser Stelle soll noch bemerkt werden, daß durch Einlöten von Drahtbrücken bezüglich der HOLD- und READY-Signale, zwei verschiedene Arbeitsweisen des Mikroprozessors gewählt werden können. Entsprechende Hinweise werden in Kapitel 4 gegeben.

3.11. System-Steuer-Baustein SAB 8228

Der SAB 8228-System-Steuer-Baustein erzeugt die Steuer-Bus-Signale, die ihrerseits die Prozessorperipherie (Speicher- und Ein-/Ausgabe-Bausteine) leiten. Sie sind dem Anwendersystem wie folgt zugänglich:

- $\overline{\text{I/O W}}$: Anschluß E
- $\overline{\text{I/O R}}$: Anschluß L
- $\overline{\text{MEMW}}$: Anschluß U
- $\overline{\text{MEMR}}$: Anschluß T

Der System-Steuer-Baustein puffert daneben auch den Daten-Bus.

3.12. Programm-Unterbrechungen („Interrupt")

Eine Ein-Ebenen-Programm-Unterbrechung erfolgt, wenn Anschluß H (INT REQ) auf Masse gelegt ist. Der System-Steuerbaustein löst dann eine RESTART-Anweisung (RST 7) aus, die dem Mikroprozessor übermittelt wird. Durch diese einfache Schaltmaßnahme ist die Durchführung von Programmunterbrechungen ohne Hardware-Aufwand möglich.

3.13. Adreß-Puffer (wahlweise)

Die SAB 8212-Adreß-Puffer sind im SIKIT-DK/8080 nicht enthalten. Sie müssen eingesetzt werden, wenn mehr als 1 KByte-RAM-Speicher und mehr als 4 KByte-ROM-Speicher angeschlossen werden. Ihre Tri-State-Struktur erlaubt das Sperren des Adreß-Busses über ein H-Signal an Anschluß S; ihr maximaler Ausgangsstrom für L-Signal beträgt 15 mA.

3.14. SIKIT-DK/8080-Speicherstruktur

Der Speicher ist unterteilt in einen Bereich für Festspeicher (PROM's, ROM's) und Schreib-/Lese-Speicher (RAM's); vgl. Bild 17.

Bild 17
Raster=Speicheraufteilung des SIKIT-DK/8080

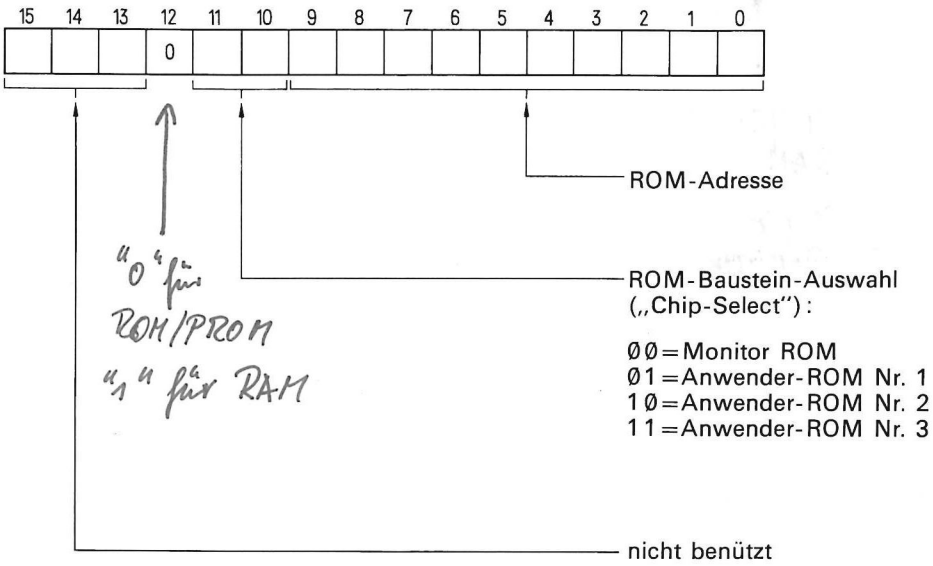
3.14.1. ROM-Dekoder und Befehlsspeicher

Maximal können auf der Platine des SIKIT-DK/8080 vier Bausteine SAB 8708 oder SAB 8308 untergebracht werden, wobei das mit dem System-Monitor programmierte ROM in Position A14 einzusetzen ist.

Der SAB 8205-ROM-Dekoder dient zur Aktivierung des adressierten ROM-Schaltkreises. Bild 18 zeigt das ROM-Adreß-Format.

Beachten Sie, daß der Inhalt des Programm-Speicherbereichs nicht durch Computeranweisungen, sondern nur durch Auswechseln der programmierten Bausteine geändert werden kann.

Bild 18
ROM/PROM-Adreß-Format

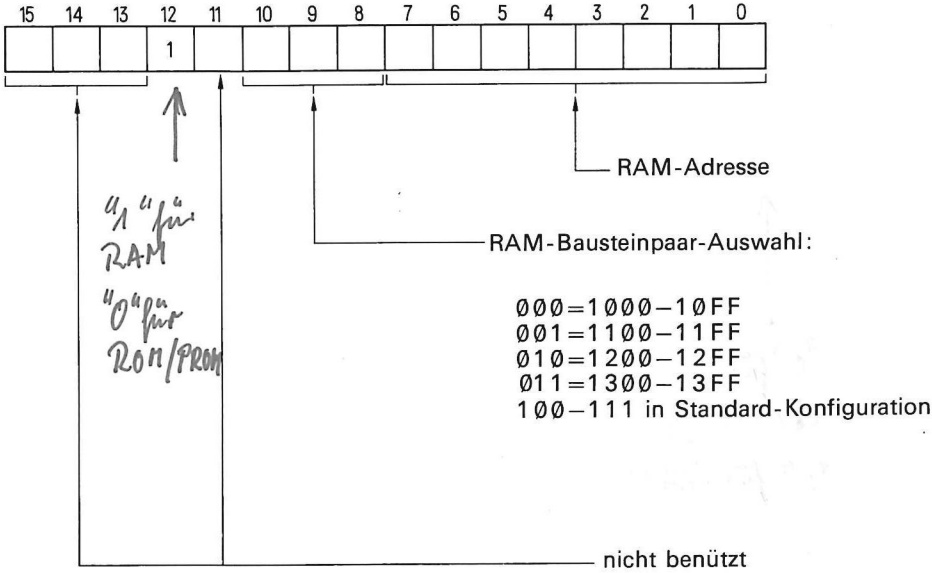


3.14.2. RAM-Dekoder und Datenspeicher

Der maximale Schreib/Lesespeicherausbau auf der SIKIT-DK/8080-Platine beträgt 1 KByte entsprechend 8 Bausteinen SAB 8111-1, von denen 2 bei Lieferung des Kits standardmäßig mitgesandt werden. Der 8205-RAM-Dekoder wählt das adressierte RAM-Baustein-Paar aus; Bild 19 zeigt das entsprechende RAM-Adreß-Format.

Sämtliche RAM's sind gesperrt, solange der Takt-Generator ein RESET-Signal liefert.

Bild 19
RAM-Adreß-Format



3.15. Die Schaltung zur Festlegung der Übertragungsgeschwindigkeit

liefert den Sende- und Empfangstakt für den Baustein SAB 8251. Sie besteht aus den drei integrierten Schaltungen: 1 × 74S161 und 2 × FLJ411.

Sie übernimmt das 18,432 MHz-OSC-Signal vom Takt-Generator und leitet daraus Signale für Übertragungsgeschwindigkeiten zwischen 75 und 4800 Baud ab. Die gewünschte Übertragungs-Rate kann durch Drahtbrücken festgelegt werden (Näheres siehe Kapitel 4).

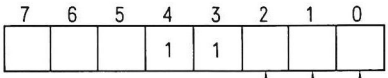
3.16. Serienschnittstelle mit SAB 8251

Der Baustein SAB 8251 ist ein universeller Synchron-/Asynchron-Sender/Empfangs-(„USART"-)Baustein. Er übernimmt die parallelen Informationen vom Daten-Bus und gibt sie seriell an die Peripherie weiter. Ebenso kann er serielle Daten von einem Peripheriegerät empfangen und sie über den 8-Bit-Daten-Bus, dem Mikroprozessor in paralleler Form zur Verfügung stellen. Bild 20 zeigt das zugehörige Adreßformat.

Bild 20
Serienschnittstellen-Adreß-Format

FA hex. Serienschn. Daten

FB hex. Serienschnittst. Steuerung



Steuerung/Daten („Control/Data")
Ø = Daten
1 = Steuerung

nicht benutzt

SAB 8251 Baustein-Auswahl:
Ø = angesprochen
1 = nicht angesprochen

2 Stopbits
keine Parität
8 Datenbit's
Baud-Rate-Faktor 64

Die Baud-Rate, mit der der Baustein SAB 8251 die Daten übermittelt und empfängt, wird vom vorher beschriebenen Teiler (Punkt 3.14.) abgeleitet. Durch Drahtverbindungen kann eine der folgenden drei Pegelarten ausgewählt werden, die in Kapitel 4 noch näher erläutert werden:

- RS-232-Pegel für Datensichtgeräte
- Stromprägungsschaltung für BEA 3913 (TTY)
- TTL-Pegel

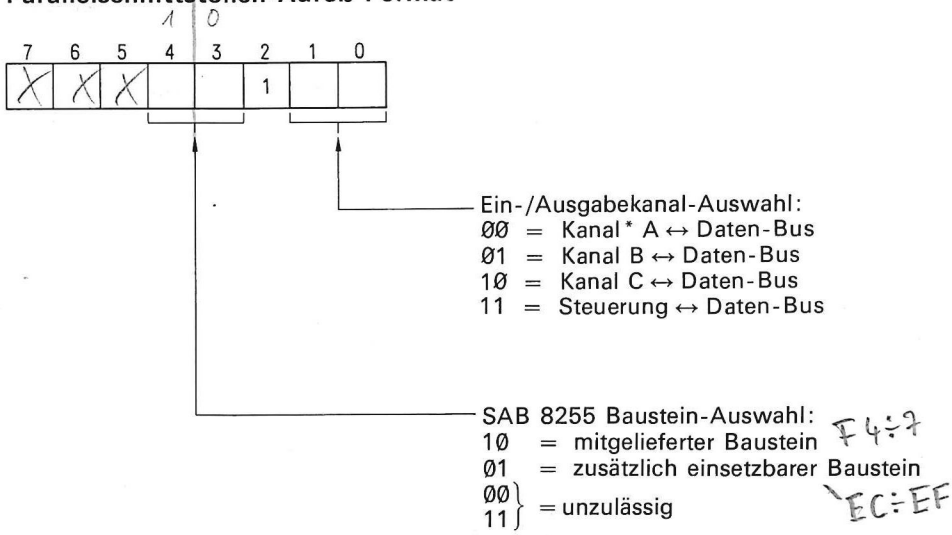
3.17. Übrige Ein-/Ausgabeleitungen

Der maximale Ausbau der SIKIT-DK/8080-Platine umfaßt 48 E/A-Leitungen, die über zwei Bausteine SAB 8255 gesteuert werden. Ein Baustein SAB 8255 ist serienmäßig im Kit enthalten.

Jeder Baustein SAB 8255 enthält drei 8-Bit-Kanäle, die in ihrer Funktion wahlweise als Ein- bzw. Ausgabekanäle verwendbar sind.

Das entsprechende Adreß-Format ist in Bild 21 dargestellt.

Bild 21
Parallelschnittstellen-Adreß-Format



* = „Port"

Es soll hier noch darauf hingewiesen werden, daß die Kanal-Ausgänge nicht weiter verdrahtet sind, um den Anwender in der Flexibilität des SIKIT-DK/8080-Einsatzes nicht zu beschränken.

Die Anschlüsse müssen daher gesondert verdrahtet werden.

4. Endzusammenbau und Überprüfung

4.1. Allgemeines

Sie haben an dieser Stelle Ihre Platine bis auf die Mikrocomputer-Bausteine fertig aufgebaut und sich über die Grundzüge der Mikrocomputer-Funktion informiert. Sie können nun die Platinen-Montage zu Ende führen und ihre Funktion überprüfen.

4.2. Spezifische Anpassung des Computer-Systems

SIKIT-DK/8080 wurde so entwickelt, daß er durch Einfügen von Drahtbrücken an Ihre speziellen Erfordernisse angepaßt werden kann.

Die nachfolgenden Fragen sollen Ihnen bei der Klärung Ihrer spezifischen Anpassungsforderung helfen.

1. Muß der Mikroprozessor HOLD- oder WAIT-Status einnehmen können?
2. Müssen die System-Busse gesperrt werden können?
3. Welches Eingabegerät möchten Sie zur Kommunikation mit Ihrem Computersystem verwenden? (z.B. Konsole BEA 3913, Bildschirmgerät)?
4. Wie hoch ist die Datenübertragungs-Rate dieses Gerätes?
5. Möchten Sie die Adreß-Leitungen mit SAB 8212-Schaltkreisen zwischenpuffern?
6. In welcher Weise und was für Information sollen zwischen Ihrem Computersystem und dessen Peripherie übertragen werden?

Bitte klären Sie zunächst diese Frage anhand Ihrer Systemanalyse und des Mikroprozessors-Datenbuchs und beginnen Sie dann, die im folgenden bezeichneten Drahtbrücken (ggf. isoliert!!) einzufügen.

4.2.1. Zu Frage 1:

HOLD- und WAIT-Möglichkeiten

- Um den HOLD-Status zu sperren, verbinden Sie die Punkte J5–2 und J5–3.
- Um den HOLD-Status freizugeben, verbinden Sie die Punkte J5–1 und J5–2.
- Wenn das Signal READY einen 8080-WAIT-Status herbeiführen soll, verbinden Sie die Punkte J5–8 und J5–9. Wenn nicht, verbinden Sie J5–8 und J5–7.

4.2.2. Zu Frage 2:
Freigabe der Busse

- Sollten die Busse extern gesperrt werden (z.B. für direkten Speicherzugriff), verbinden Sie Punkt J5–5 mit J5–6.
- Sollten die Busse immer unter der internen Computerkontrolle bleiben, verbinden Sie Punkt J5–4 mit J5–5.

4.2.3. Zu Frage 3:
Festlegung der Datenübertragungsrate und des Serienschnittstellenpegels

- Für 110 Baud, die Standard-Übertragungsrate für BEA 3913, verbinden Sie Anschluß 4 mit Anschluß 5.
- Zur Wahl einer anderen Übertragungsrate verbinden Sie den Anschluß 29 mit einem der Anschlüsse 31 bis 37, nach der folgenden Auswahltabelle.

Anschlußtabelle der Baud-Rate

Baud-Rate	Anschluß 29 mit
4800	31
2400	32
1200	33
600	34
300	35
150	36
75 oder 110	37

4.2.4. Zu Frage 4:
Festlegung des Serienschnittstellenpegels

Der Pegel der Serienschnittstelle (Anschluß J3) läßt sich ebenfalls durch Drahtverbindungen definieren.

Drei Schnittstellen-Typen stehen zur Verfügung:

- TTL-Standard-Schnittstelle
- 3913-Konsolen-Schnittstelle
- Sichtgeräte-Schnittstelle

Je nach gewünschter Schnittstelle sind die in der Anschlußtabelle für Datenübertragungsstufen aufgeführten Verbindungen herzustellen. Außerdem müssen die Anschlüsse A und B miteinander verbunden werden, wenn Ihr System nicht mit einem Modem arbeitet.

Anschlußtabelle für Datenübertragungsstufe

CRT Configuration Anschlüsse	TTY Configuration Anschlüsse	TTL Configuration Anschlüsse
23 mit 24	23 mit 26	23 mit 25
17 mit 18	18 mit 19	17 mit 18
9 mit 10	10 mit 11	12 mit 13
13 mit 14	13 mit 14	2 mit 3
2 mit 3	1 mit 2	20 mit 21
6 mit 8	7 mit 8	–
27 mit 28	15 mit 16	–
21 mit 22	21 mit 22	–

4.2.5. Zu Frage 5: Adreß-Bus-Puffer

Falls Sie keine SAB 8212-Pufferschaltkreise benötigen, müssen jeweils in Schaltkreisposition A11 **UND** A12 die folgenden Schaltkreisanschlüsse **auf der Unterseite** (nichtbedruckte Seite) verbunden werden.

- Anschluß 3 mit 4
- Anschluß 5 mit 6
- Anschluß 7 mit 8
- Anschluß 9 mit 10
- Anschluß 15 mit 16
- Anschluß 17 mit 18
- Anschluß 19 mit 20
- Anschluß 21 mit 22

4.2.6. Zu Frage 6: Anschluß der Ein-/Ausgabeleitungen

Der Datenverkehr des Computersystem SIKIT-DK/8080, wird über den Steckverbinder J3 durchgeführt. (Anschlußzuordnung siehe Tabelle.)

Anschlußzuordnung für Datenübertragungs-Schnittstelle

J3 Pin	CRT Configuration	TTY Configuration	TTL Configuration
1			
2	CRT REC. DATA		TTL REC. DATA
3	CRT XMIT DATA		TTL XMIT DATA
4			
5	+12 V (DC)		
6			
7	SIGNAL GND		SIGNAL GND
8	+12 V (DC)		
9			
10			
11			
12		TTY REC. RETURN	
13		TTY XMIT	
14			
15			
16			
17			
18			
19			
20	+12 V (DC)		
21			
22			
23			
24		TTY REC.	
25		TTY XMIT RETURN	

Alle restlichen Anschlüsse, z.B. die des Ein-/Ausgabe-Bausteins SAB 8255, können vom Anwender völlig wahlfrei belegt werden; zu den beiden Steckverbindern passende Anschlußstecker sind ebenfalls im Bausatz enthalten.

Die mit Buchstaben gekennzeichneten Platinenanschlüsse führen die Kontrollsignale des Computersystems. Ihre Zuordnung und Bedeutung entnehmen Sie nachfolgender Tabelle.

SIKIT-DK/8080-Steuer-Bus-Anschlüsse

Anschluß	Mnemonic	Bedeutung
A	$\overline{\text{CTS}}$	Sendebereitschaft
B	$\overline{\text{RTS}}$	Sendeaufforderung
C	Ø2 (TTL)	2,048 MHz-Takt
D	$\overline{\text{DSR}}$	Datenübertragungseinrichtung – Bereit
E	$\overline{\text{I/O W}}$	Ein-/Ausgabe-Schreiben
F	$\overline{\text{DTR}}$	Datenstation–Bereit
H	$\overline{\text{INT REQ}}$	Unterbrechungs-Anforderung
I	$\overline{\text{STATUS STROBE}}$	Zustandssignal auf dem Daten-Bus
K	$\overline{\text{OSC}}$	18,432 MHz-Oszillator
L	$\overline{\text{I/O R}}$	Ein-/Ausgabe-Lesen
M	$\overline{\text{HLDA}}$	Quittungs-Signal für Halt-Befehl
N	$\overline{\text{INTA}}$	Quittungs-Signal für Unterbrechung
P	READY	Bereit
R	$\overline{\text{HOLD}}$	Halt
S	$\overline{\text{SYSTEM BUS ENABLE}}$	Daten- und Adreß-Bus – Freigabe
T	$\overline{\text{MEMR}}$	Speicher Lesen
U	$\overline{\text{MEMW}}$	Speicher Schreiben
V	RESET	Rücksetzen

4.3. Einfügen der Mikrocomputer-IS

4.3.1. Einbauhinweise

Bitte beachten Sie!
Obwohl die MOS-Bausteine (8080, 8111-1, 8251, 8255, 8708) sämtlich Eingangsschutzstrukturen eingebaut haben, sollte man sie unbedingt vor statischer Elektrizität schützen. Die folgenden Vorsichtsmaßregeln sind daher unbedingt zu beachten.

- Alle Gegenstände, die mit der Schaltung in Berührung kommen, sollten auf gleichem Potential liegen; denken Sie dabei an Lötkolben, Zinn, Werkzeuge usw.
  Daher ist häufiger körperlicher (möglichst elektrischer) Kontakt zwischen Arbeitstisch, Schaltung und den genannten Gegenständen wichtig (z.B. über leitende Schaumstoff-Unterlage).
- Gewöhnen Sie sich an, die IS nur in leitender Umgebung zu berühren und zu lagern.
- Ebenso sollten Sie auch die Leiterplatte berühren, bevor Sie einen IS einsetzen.
- Fassen Sie den IS-Baustein möglichst nur am Gehäuse und nicht an seinen Anschlüssen an.
- Berühren Sie grundsätzlich einen MOS-Baustein nur mit Gegenständen (z.B. Werkzeugen), die Sie vorher mit dem Arbeitstisch in Berührung gebracht haben.

4.3.2. Ausrichten der IS-Anschlüsse

Falls Anschlüsse eines IS verbogen sind, sollten Sie unter keinen Umständen versuchen, das Bauteil mit Gewalt in seine Fassung einzusetzen, sondern erst alle Pins ausrichten. Dies erreicht man einfach, indem man das IS-Gehäuse zwischen die Fingerspitzen nimmt und die Beinchen durch leichten Druck auf das gesamte Gehäuse gegen den Arbeitstisch so richtet, daß sie alle senkrecht zum Gehäuse stehen.

4.3.3. Einsetzen der IS in die Computerplatine

1. Jedes IS-Gehäuse trägt eine Markierung (Aussparung, Punkt, Pluszeichen oder die Ziffer „1"), die eindeutig die Lage der Anschlüsse definiert. Es ist besonders darauf zu achten, daß die IS richtig herum eingesetzt werden, weil bei falscher Orientierung eine Zerstörung des IS bei Inbetriebnahme zu befürchten ist. Die obengenannte Markierung muß jeweils neben der auf der Platine aufgedruckten Ziffer „1" zu liegen kommen.
2. Stecken Sie die Pins auf einer Seite des IS in ihre hierfür bestimmten Bohrungen bzw. Fassungen auf der bedruckten Platinenseite. DRÜCKEN SIE DIE PINS NICHT GANZ HINEIN! Sollten Sie Schwierigkeiten haben, die Pins in die Löcher zu bringen, helfen Sie mit der Klinge eines kleinen Schraubenziehers nach.
3. Führen Sie die Pins der anderen IS-Seite genauso in die zugehörigen Löcher ein. Sobald alle Pins an ihrer Stelle sind, wird der IS auf seinen Platz gedrückt, indem er ganz leicht hin- und herbewegt wird, bis er am Anschlag auf der Platine oder in der Fassung sitzt.
4. IS-Bausteine, die nicht in Fassungen eingesetzt werden, müssen nun noch von der Rückseite verlötet werden (keine Lötstelle vergessen)!

4.3.4. Herausnehmen von IS aus ihren Fassungen

Muß ein IS aus seiner Fassung wieder entfernt werden, bewegt man ihn leicht hin und her, bis er sich herauslösen läßt. Sobald eine Lücke zwischen IS und Fassung entstanden ist, kann man mit einem Schraubenzieher nachhelfen. Versuchen Sie, während des ganzen Vorgangs den IS parallel zur Fassung zu halten.

4.3.5. Einsetzen des Taktgenerators

Setzen Sie den SAB 8224 (Taktgenerator) auf seine Position A8 (Bild 22). Gehen Sie dabei bitte besonders sorgfältig vor, da dieser Baustein (neben dem Prozessor selbst) eines der empfindlichsten Bauelemente darstellt. Sie können jetzt bereits Stromversorgung, Systemtakt und die Funktion des RESET-Schalters prüfen.

Führen Sie diesen Test bitte folgendermaßen durch:

- Legen Sie die Speisespannungen an die Anschlußösen E1–E6 der Platine. Falls Sie über ein SME-Gerät verfügen, kann dies wegen der Kompatibilität beider Geräte einfach durch Einstecken der SIKIT-DK/8080-Platine in das SME-Chassis erfolgen.
- Schalten Sie die Spannungen ein.
- Prüfen Sie mit dem Spannungsmeßgerät die +5 V, +12 V, −10 V und −5 V Gleichspannungen an ihren Anschlüssen auf der SIKIT-DK/8080-Platine.
- Drücken Sie mehrmals die RESET-Taste und prüfen Sie, ob beim SAB 8224 an Pin 1, +4 V liegt (RESET-Signal).

Anmerkung:
Gewöhnen Sie sich an, die RESET-Taste jedesmal nach Einschalten der Spannungsversorgung zu betätigen.

- Wenn Sie einen Oszillographen haben, prüfen Sie, ob beim SAB 8224 an den Pins 10 und 11 jeweils der 2,048 MHz-Takt anliegt ($\Phi 2$ und $\Phi 1$ nacheinander).
- An Pin 12 muß ein 18,432 MHz-Takt liegen (Signal OSC).
- Der erste Test ist damit beendet. Schalten Sie den Strom ab.

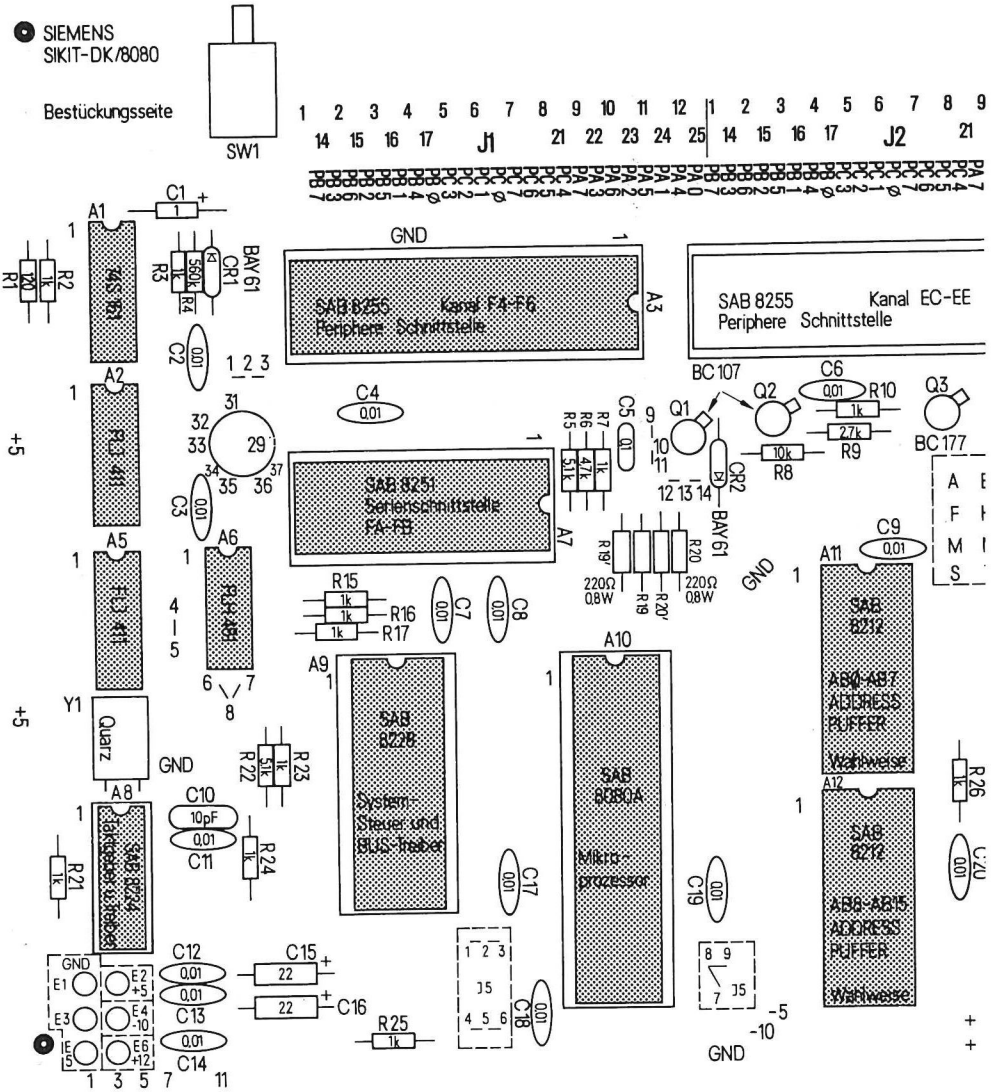
4.3.6. Einsetzen der übrigen IS-Bausteine (Siehe Bild 22)

Nachdem Sie sich davon überzeugt haben, daß der SIKIT-DK/8080 korrekt mit Strom versorgt wird und die System-Takte und das RESET-Signal überprüft sind, können Sie den restlichen Einbau der Bausteine vornehmen.

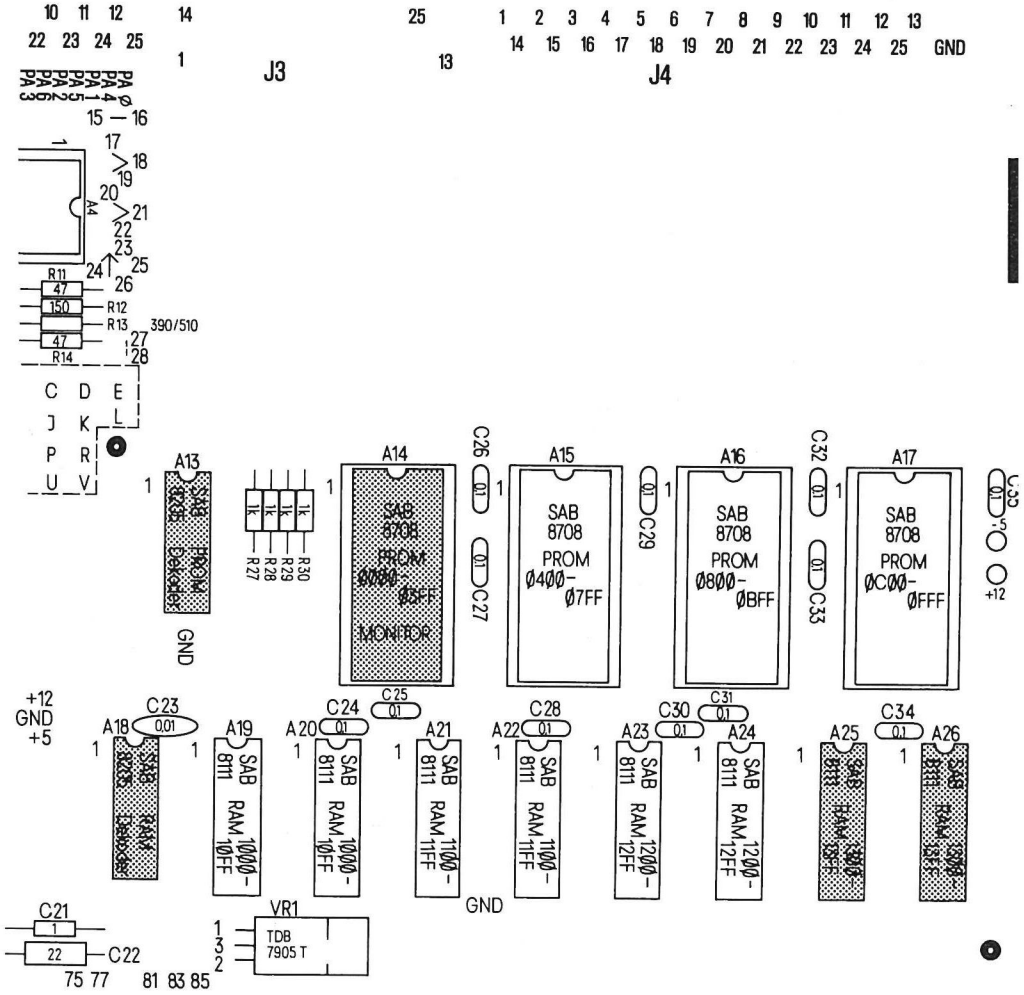
Dies geht folgendermaßen vor sich:

- Löten Sie den 74S161 auf Position A1.
- Löten Sie die FLJ411 auf Pos. A2 und A5.
- Löten Sie den FLH481 auf Pos. A6.
- Falls Adreßpuffer verwendet werden, löten Sie die SAB 8212 auf Pos. A11 und A12.
- Löten Sie die SAB 8205 auf Pos. A13 und A18.
- Löten Sie die SAB 8111-1 auf Pos. A25 und A26.
- Setzen Sie den SAB 8228 in die Fassung A9 ein.
- Setzen Sie den SAB 8080A in die Fassung A10 ein.
- Setzen Sie den SAB 8251 in die Fassung A7 ein.
- Setzen Sie den SAB 8255 in die Fassung A3 ein.
- Setzen Sie das bereits programmierte Monitor-ROM-„SDK" in seine Fassung A14 ein.

Bild 22
Einlöten bzw. Einsetzen der IS



Endzusammenbau und Überprüfung



4.4. Inbetriebnahme

Ihr Computersystem SIKIT-DK/8080 ist nun komplett aufgebaut. Sie brauchen es nur noch in Betrieb zu nehmen!
Das geschieht folgendermaßen:

- Schließen Sie Ihre System-Konsole (BEA 3913 oder Sichtgerät) an den Steckverbinder J3 an.
- Schalten Sie die Stromversorgung für Ihren SIKIT-DK/8080-Computer und Ihre Konsole ein.
- Drücken Sie die RESET-Taste.

Der Monitor quittiert Ihre Aktion nun mit

MCS-80 KIT

Jetzt sind Sie zum Produktiveinsatz des SIKIT-DK/8080 bereit!

4.5. Hinweise zur Fehlersuche

Sollte Ihr erster Inbetriebnahmeversuch nicht ganz so erbaulich verlaufen sein, schalten Sie bitte den Strom aus und führen Sie folgende Kontrollen durch:

1. Prüfen Sie, ob alle Widerstände richtig eingebaut sind. Den Gesamtbestückungsplan finden Sie in Kapitel 8.
2. Stellen Sie fest, ob alle Kondensatoren richtig (Polarität!) eingebaut wurden.
3. Überprüfen Sie die Polarität der beiden Dioden CR1 und CR2.
4. Überzeugen Sie sich, daß die Nasen der drei Transistoren in der richtigen Position sind.
5. Prüfen Sie, ob alle IS richtig eingebaut sind.
6. Vergewissern Sie sich über die Richtigkeit und Vollständigkeit sämtlicher Drahtbrücken.
7. Falls alle diese Bemühungen keinen Fehler zutage fördern, wenden Sie sich an Ihre zuständige Vertriebsstelle. (Verzeichnis am Schluß des Buches.)

5. Das MONITOR-Programm

5.1. Allgemeines

Das im ROM „SDK" bereits untergebrachte MONITOR-Programm (1 KByte in Speicherstelle Ø...1Ø23) erleichtert Ihnen die Inbetriebnahme Ihres Computers durch seine Möglichkeiten zur Fehlersuche und Korrektur, Programmaufruf und Abwicklung des Datenverkehrs mit Konsole und Peripherie.

Im einzelnen versteht er über die Konsole (Bedienungsblattschreiber oder Datensichtgerät) einzugebende Befehle für folgende Funktionen:

- Anzeige von Speicherinhalten
- Änderung von Speicher- und Registerinhalten
- Start von Anwenderprogrammen
- Einfügen von Aufsetzpunkten („Break Points") im Anwenderprogramm
- Weitergabe von hexadezimalen Eingabedaten von der Konsole an den Arbeitsspeicher.

Darüber hinaus ermöglicht der Monitor, wie angedeutet, den Dialog zwischen dem SIKIT-DK/8080-Computer und dem Anwender über die Systemkonsole.

Eingaben des Benutzers, die den Anforderungen der im folgenden beschriebenen Kommandosprache genügen müssen, quittiert er durch die gewünschten Aktionen oder aber durch Ausgabe eines Kommentars.

Nach dem Start meldet sich der Monitor durch Ausgabe eines Punktes („.") und wartet auf das erste Benutzerkommando.

5.2. Funktionsweise

MCS 80 KIT
bzw. CMD:

Der Monitor meldet sich, wie bereits erwähnt, mit dem Zeichen „Punkt" („.") und erwartet dann die Eingabe eines Kommandos. Jedes dieser Kommandos besteht aus einem Buchstaben zur Identifikation der Kommandoart und einer nachstehenden Folge von Parametern.

Es gibt numerische und alphabetische Parameter, wobei numerische Parameter im hexadezimalen Format angegeben werden (Ø...9 und A, B, C, D, E und F).

Maximal 4 hexadezimale Zeichen werden vom Monitor erkannt. Bei längeren Zeichenfolgen werden einfach nur die niederwertigen 4 Zeichen interpretiert.

5.2.1. Programmerstellung und -Test mit Hilfe des Monitors

Mit dem Monitor können kleinere Programme gestartet, getestet und ausgeführt werden. Hierfür stehen Möglichkeiten zur Mikroprozessor-Register-Anzeige und Inhaltsänderung zur Verfügung, außerdem zum Programmladen, Erkennen der RST 7-Anweisung und zum unbedingten Sprung zur RAM-Adresse 13 FDH. (Durch software- oder hardware-mäßige Festlegung der RST 7-Anweisung läßt sich die Fortsetzung des Programms an einer vorbestimmten Stelle bei Verarbeitung dieser Anweisung erzwingen.)

Zum Rücksprung auf dem Monitor sollte die RST 1-Anweisung (hard- oder software-mäßig vorgegeben) Verwendung finden.

Vom Monitor werden bei Ausführung dieser Anweisung automatisch sämtliche Mikroprozessor-Register incl. Status-Bits, Programmzähler und Stapelzeiger „gerettet"; sie können durch das Kommando „X" abgefragt und durch „G" dem Prozessor vorgegeben werden.

5.2.2. Ein-/Ausgabeteil

Über zwei Routinen erfolgt die Ein- bzw. Ausgabe jeweils eines Zeichens von/an der/die Systemkonsole.

Bitte beachten Sie, daß mit „bit Ø" das niederwertigste (d.h. am weitesten rechts stehende Bit) bezeichnet wird und daß der interne Code der 7-Bit-ASCII-Code ist (ohne Paritätsbit).

5.3. Kommando-Formate

An das Format der Kommandoeingaben werden folgende formale Bedingungen gestellt:

- Jedes Kommando wird durch Eingabe eines „Wagenrücklauf"-(=„Carriage Return"-)Zeichens abgeschlossen; mit Ausnahme der Kommandos „S" und „X" wird das Kommando erst nach Eingabe dieses Wagenrücklaufzeichens akzeptiert. Dadurch ist eine nach dem Eintippen bis zum Drücken der „Wagenrücklauf"-Taste noch evtl. nötige Korrektur (etwa durch die RUB OUT-Taste) möglich.
- Falls nicht anders vermerkt, kann entweder das Leerzeichen („␣") oder ein Komma („,") als Trennzeichen verwendet werden. Mehrere aufeinanderfolgende Leerzeichen oder Kommata werden als Null-Parameter interpretiert (nur bei Kommando „X" zulässig!).

5.3.1. Anzeige des Speicherinhalts: Kommando „D"

Format

D *von-adresse, bis-adresse*

Ergebnis des Kommandos ist Ausgabe der Speicherinhalte von *„von-adresse"* bis *„bis-adresse"* auf der Konsole.

In jeder Zeile der Ausgabe stehen die erste in dieser Zeile angezeigte Speicheradresse (=4 hexadezimale Zeichen) und die Inhalte von maximal 16 Speicherstellen, die durch je 2 hexadezimale Zeichen dargestellt sind.

Durch Drücken der „ESCAPE"-Taste (ESC) auf der Konsole kann der Monitor zum Abbruch der Ausgabe veranlaßt werden.

Beispiel für die Ausgabe:

```
    D9,2A
0009 00 11 22 33 44 55 66
0010 77 88 99 AA BB CC DD EE FF 10 20 30 40 50 60 70
0020 80 90 A0 B0 C0 D0 E0 F0 01 02 03
```

5.3.2. Programmstart: Kommando „G"

Format:

G *startadresse*

Angabe von *„startadresse"* ist nicht zwingend; wird *„startadresse"* nicht angegeben, verwendet der Monitor hierfür die oberste Adresse im Stapelspeicher.

Wirkung des Kommandos:
Die Programmkontrolle des Mikroprozessors geht vom Monitor auf das Anwenderprogramm über, d.h. der SIKIT-DK/8080-Computer führt als nächstes die Anweisung durch, die er an der Speicherstelle *„startadresse"* findet (*„startadresse"* sollte eine RAM-Adresse sein, in die ein Befehl des Anwenderprogramms eingeschrieben ist).

Beispiel für die Monitormeldung:

Eingabe: G14ØØ
Wirkung: Die Programmsteuerung geht an Speicherstelle 14ØØH über.

5.3.3. Einschreiben eines Befehls in den RAM-Bereich: Kommando „I"

Format

I *adresse*

Mit dem „I"-Kommando lassen sich einzelne Anweisungen oder auch ganze Anwenderprogramme ins RAM einschreiben. Nach Eingabe des „I" und des „Wagenrücklauf"-Zeichens erwartet der Monitor die Eingabe einer hexadezimalen Zeichenfolge, die er in Binärwerte umsetzt und ab der Speicherstelle „adresse" in den RAM-Speicherbereich einschreibt (2 hexadezimale Zeichen entsprechen dem Inhalt einer Speicherzelle =1 Byte!).

Auf diese Weise können (im Rahmen der verfügbaren Speicherkapazität) beliebig viele Speicherstellen-Inhalte eingegeben werden; die Eingabe wird durch Drücken der Tasten ESC oder ALTMODE beendet, was vom Monitor mit Ausgabe des Dollarzeichens („$") quittiert wird.

Auf die Eingabe einer ungeraden Anzahl hexadezimaler Zeichen weist der Monitor durch Ausgabe einer Null („Ø") hin; die Eingabe von Leerzeichen („␣"), Kommata („,") und Wagenrücklauf wird ignoriert.

Erstes Beispiel einer Eingabe:

I141Ø
112233445566778Ø99$

Das Beispiel veranlaßt folgende RAM-Belegung:

141Ø 11 22 33 44 55 66 77 88 99

Zweites Beispiel:

I144Ø
123456789$

Das zweite Beispiel veranlaßt folgende RAM-Belegung:

144Ø 12 34 56 78 9Ø

wobei, wie erwähnt, an das letzte Zeichen eine Ø angefügt wurde, da eine ungerade Anzahl hexadezimaler Zeichen eingegeben würde, wodurch die Speicherstellenbelegung „nicht aufgeht".

5.3.4. Datentransfer im Speicher: Kommando „M"

Format

M *von-adresse, bis-adresse, ziel*

Das Kommando überträgt die Inhalte der Speicheradressen *„von-adresse"* bis *„bis-adresse"* einschließlich in den Speicherbereich, der mit der Adresse *„ziel"* beginnt. Soweit sich die beiden Speicherbereiche nicht überlappen, bleibt der „Absendespeicherbereich" unverändert.

Das Kommando „M" wird Byte-weise ausgeführt, wobei das Byte der *„von-adresse"* als erstes übertragen wird. Problematisch ist es, wenn *„ziel"* im Speicherbereich zwischen *„von-adresse"* und *„bis-adresse"* liegt.

Dann liefert beispielsweise das Kommando

M141Ø, 141F, 1411

den Inhalt „1A1A1A..." an die Speicherstellen 141Ø bis 142Ø, wenn in 141Ø „1A2" (hexadezimal) enthalten war. In diesem Fall überträgt der Monitor dauernd Daten, bis das Sendefeld abgearbeitet oder die Adresse ØFFFFH erreicht ist.

Beispiel: Durch

M141Ø, 15ØF, 151Ø

werden 256 Byte von den Adressen 141Ø bis 15ØF an den Speicherbereich151Ø–16ØF übertragen.

5.3.5. Änderung einer Speicherstelle: Kommando „S"

Format

S *adresse*

Durch dieses Kommando ist die Prüfung und (falls gewünscht) das Ändern des Inhaltes der Speicherstelle *„adresse"* möglich.

Das Kommando wird wie folgt behandelt:

- Tippen Sie den Buchstaben „S", die hexadezimale Adresse der ersten anzuzeigenden Speicherstelle und dann ein Leerzeichen („␣") oder ein Komma („,").
- Der Monitor veranlaßt nun die Anzeige des Inhalts der Speicherstelle und die Ausgabe eines Gedankenstrichs („–").
- Soll der Speicherstelleninhalt geändert werden, können Sie jetzt den neuen Wert eingeben (2 hexadezimale Zeichen), danach ein Leerzeichen („␣"), Komma („,") oder Wagenrücklauf.
- Soll der Speicherstelleninhalt nicht geändert werden, wird der Wert weggelassen und lediglich Leerzeichen, Komma oder Wagenrücklauf gegeben.
- In beiden Fällen schaltet der Monitor durch das Leerzeichen oder Komma zur nächstfolgenden Speicherstelle weiter.
- Bei Wagenrücklauf wird die Ausführung des „S"-Kommandos abgebrochen.

Beispiel: Durch

S145Ø AA– BB–CC Ø1–13 23–24

wird Speicherstelle 145Ø nicht geändert (Inhalt ist und bleibt AA). 1451 wird von BB auf CC umgeschrieben, 1452 (Inhalt vorher Ø1) wird auf 13 gesetzt und 1453 (vorher 23) bekommt den Inhalt 24.

5.3.6. Ausgabe und Änderung der Prozessor-Register: Kommando „X"

Format:

X *reg*

Die Angabe von *„reg"* ist wahlweise.

Wird kein Register explizit durch *„reg"* identifiziert, werden sämtliche Registerinhalte angezeigt.

Anzeige "Carry" nicht möglich (Fehler im Monitor-Programm "Restart Entry Point" Nur mit korrigierten UP möglich

Für *„reg"* steht die 8080-Registerbezeichnung

A = Akkumulator,
B, C, D, E, F = Flags (= Statusbits),
H, L, M (= Inhalt von H **und** L),
P (= Befehlszähler) und
S (= Stapelzeiger).

Die Verwendung des Kommandos geschieht folgendermaßen:

- Drücken Sie die Tasten X, geben Sie die gewünschte Registerbezeichnung ein und/oder das „Wagenrücklauf"-Zeichen.
- Für jedes Register werden nun 2 hexadezimale Zeichen (≙ 8 Bits) ausgegeben; danach folgt ein Gedankenstrich („–").
- Der Registerinhalt kann nun durch Angabe eines gewünschten Werts verändert werden, worauf wieder ein Leerzeichen, Komma oder Wagenrücklauf-Zeichen zu folgen hat.
- Soll der Registerinhalt unverändert bleiben, ist nur ein Leerzeichen, Komma oder Wagenrücklauf-Zeichen zu geben.
- Bei Eingabe von Komma oder Leerzeichen geht der Monitor automatisch zur Anzeige des nächsten Registers in alphabetischer Reihenfolge über, bei Wagenrücklauf wird die Bearbeitung des X-Kommandos abgebrochen.

Beispiel:

```
XA   AA–  BB–  CC–  DD–  EE–  FF–  12–  34–  1234–  0000
XA   AA–  23–  CC–  01–  EE–  FF–  12–  34–  1234–  1010
X
A–AA B–23 C–CC D–01 E–EE F–FF H–12 L–34 M–1234 P–01CF S–03CD
```

5.4. Ein-/Ausgabe

Der Verkehr des Monitors mit der Konsole läuft über den Serienschnittstellen (USART)-Baustein SAB 8251 ab, der beim Zusammenbau des SIKIT-DK/8080 bereits für den Verkehr mit der angeschlossenen Konsole eingestellt wurde.

Die Eingaberoutine und die Ausgaberoutine, über die dieser Verkehr software-mäßig erfolgt, können auch vom Anwender durch Aufruf zur Ein- bzw. Ausgabe in seinem Anwenderprogramm benützt werden.

5.4.1. Die Eingaberoutine CI (Konsolen-Eingabe)

überträgt ein Zeichen von der Konsole in den Prozessor-Akkumulator (= „Register A"); ihre Beginnadresse ist 3FDH. (Zur Beachtung: Hierbei werden auch die Status-Bits über Register A angesprochen!)

Beispiel:

```
CI   EQU 3FDH
     ...
     CALL CI
     STA DATA
     ...
```

5.4.2. Die Ausgaberoutine CO (Konsolen-Ausgabe)

überträgt ein Zeichen vom Register C an die Konsole, wobei durch Veränderung der Registerinhalte von A und C auch Status-Bits angesprochen werden. Beginnadresse is 3FAH.

Beispiel:

```
CO  EQU 3FAH
    ...
    MVI C,"."
    CALL CO
```

5.5. Betriebshinweise

5.5.1. Aktivierung des Programms

Nach Einschalten der Speisespannung oder RESET-Signal werden automatisch Monitor-Instruktionen ab ROM-Speicherstelle Ø zur Initialisierung durchgeführt und die Monitor-Quittung, die seine Bereitschaft zum Empfang eines Kommandos zeigt, auf der Konsole durch Ausgabe von „." angezeigt.

5.5.2. Stapelspeicherbereich (geändert je nach Version Monitorprogramme, sicherlich

Der Monitor weist seinen Stapelspeicherbereich dynamisch dem RAM-Adreßbereich 4 KByte...5 KByte zu. Die obersten 3 Bytes in diesem RAM-Bereich sind für die Sprung-Adresse reserviert, die vom Anwender festgelegt wird (=Ziel der RST 7-Instruktionen oder einer anderen hardware-mäßig zu gebenden RST-Instruktion). Dem Anwender ist der gesamte RAM-Speicherbereich mit Ausnahme des Bereichs 5 KByte – 1 KByte bis 5 KByte – 256 Byte zugänglich.

5.5.3. Testmöglichkeit durch Aufsetzpunkt

Bei Anliegen einer RST 1-Instruktion (CF hexadezimal) sichert der Monitor automatisch sämtliche Mikroprozessor-8080-Status-Informationen und es erfolgt Ausgabe der Meldung

MCS-80 KIT ('CMD:')

auf der Konsole.

Zu Testzwecken kann der Anwender nun durch das „X"-Kommando diese Statusinformation ausgeben lassen. Auf diese Weise läßt sich die Funktion des Programms gut überprüfen; Voraussetzung ist, daß im Anwenderprogramm an geeigneten „Check-Punkten" je ein RST 1-Befehl steht. Nach diesem Test und evtl. Programmänderungen wird die Ausführung des Anwenderprogramms einfach durch das Kommando „G" fortgesetzt.

5.5.4. Unterbrechungstechnik („Interrupt")

Mit dem SIKIT-DK/8080-Computer ist die Behandlung von Programmunterbrechungen möglich. Beim Anliegen einer Unterbrechungsanforderung am Mikroprozessor SAB 8080 (Eingang INT auf H-Pegel) wird automatisch eine RST 7-Anweisung ausgeführt, die zu einem unbedingten Sprung zur Speicherstelle 13FDH führt. Dadurch kann der Anwender sein Programm auch ohne Konsolen-Kommando „G" durch eine periphere Einheit (z.B. durch Schließen eines Schalters) starten.

Durch eine RST 1-Anweisung kann auch während der Programmunterbrechung jederzeit wieder der Monitor aufgerufen werden, der sich dann mit

MCS-80 KIT ('CHD')

meldet.

Ein Rücksprung zum Anwenderprogramm ist dann wieder durch das Kommando „G" möglich.

5.6. Fehlermeldungen

5.6.1. Unzulässige Zeichen

Der Monitor prüft jedes von der Konsole eingegebene Zeichen auf seine Zulässigkeit. Beim Empfang eines unzulässigen Zeichens bricht der Monitor das Kommando ab und gibt als Fehlermeldung ein „*" aus.

Beispiel 1:	D1400, 145G*
Erklärung	Das Zeichen G steht hier in einer Parameter-Liste, in der nur hexadezimale Zeichen und Trennzeichen zulässig sind.
Beispiel 2:	Y*
Erklärung	Y ist kein gültiges Kommando.

5.6.2. Adreß-Fehler

- Einige Kommandos erfordern die Angabe eines Adressenpaars in der Form
  „von-adresse, bis-adresse"
  Ist bei einem solchen Kommando der Wert von *„von-adresse"* größer oder gleich dem Wert von *„bis-adresse"*, wird das Kommando nur bezüglich der Daten in der *„von-adresse'* ausgeführt.
- Bei Eingabe einer Adresse >FFFFH werden nur die letzten 4 hexadezimalen Zeichen berücksichtigt, die übrigen ignoriert.
- Ein dritter möglicher Adreßfehler ist das Anspringen einer Speicherstelle, die in der Hardware-Konfiguration physikalisch nicht existiert. Falls ein nicht-existierender Speicherbereich als Sendebereich benutzt wurde, ist das Ergebnis dieses Kommandos nicht vorhersehbar.

  Wurde ein nicht-existenter Speicherbereich als Zielbereich angegeben, hat das Kommando keinerlei Wirkung.

blank page

```
8080 MACRO ASSEMBLER, VER 2.3  ERRORS = 0 PAGE 1

                        ;*************************************************************
                        ;
                        ;                 PROGRAM: 8080A BOARD MONITOR
                        ;
                        ;*************************************************************
                        ;
                        ;
                        ; ABSTRACT
                        ; ========
                        ;
                        ; THIS PROGRAM RUNS ON THE 8080A BOARD AND IS DESIGNED TO PROVIDE
                        ; THE USER WITH A MINIMAL MONITOR. BY USING THIS PROGRAM,
                        ; THE USER CAN EXAMINE AND CHANGE MEMORY OR CPU REGISTERS, LOAD
                        ; A PROGRAM (IN ABSOLUTE HEX) INTO RAM, AND EXECUTE INSTRUCTIONS
                        ; ALREADY IN MEMORY. THE MONITOR ALSO PROVIDES THE USER WITH
                        ; ROUTINES FOR PERFORMING CONSOLE I/O.
                        ;
                        ;
                        ;
                        ; PROGRAM ORGANIZATION
                        ; ====================
                        ;
                        ; THE LISTING IS ORGANIZED IN THE FOLLOWING WAY. FIRST THE COMMAND
                        ; RECOGNIZER, WHICH IS THE HIGHEST LEVEL ROUTINE IN THE PROGRAM.
                        ; NEXT THE ROUTINES TO IMPLEMENT THE VARIOUS COMMANDS. FINALLY,
                        ; THE UTILITY ROUTINES WHICH ACTUALLY DO THE DIRTY WORK. WITHIN
                        ; EACH SECTION, THE ROUTINES ARE ORGANIZED IN ALPHABETICAL
                        ; ORDER, BY ENTRY POINT OF THE ROUTINE.
                        ;
                        ; THIS PROGRAM EXPECTS TO RUN IN THE FIRST 1K OF ADDRESS SPACE.
                        ; IF, FOR SOME REASON, THE PROGRAM IS RE-ORG'ED, CARE SHOULD
                        ; BE TAKEN TO MAKE SURE THAT THE TRANSFER INSTRUCTIONS FOR RST 1
                        ; AND RST 7 ARE ADJUSTED APPROPRIATELY.
                        ;
                        ; THE PROGRAM ALSO EXPECTS THAT RAM LOCATIONS 5K-1 TO 5K-256,
                        ; INCLUSIVE, ARE RESERVED FOR THE PROGRAM'S OWN USE. THESE
                        ; LOCATIONS MAY BE ALTERED, HOWEVER, BY CHANGING THE EQU'ED
                        ; SYMBOL "DATA" AS DESIRED.
                        ;
                        ;
                        ; LIST OF FUNCTIONS
                        ; =================
                        ;
                        ;       GETCM
                        ;       -----
                        ;
                        ;       DCMD
                        ;       GCMD
```

gelb angestrichen: an dieser Stelle wird SP-Bereich festgelegt (hier 0x0100)

```
8080 MACRO ASSEMBLER, VER 2.3  ERRORS = 0 PAGE 2

                ;       ICMD
                ;       MCMD
                ;       SCMD
                ;       XCMD
                ;       ----
                ;
                ;       BREAK
                ;       CI
                ;       CNVBN
                ;       CO
                ;       CROUT
                ;       ECHO
                ;       ERROR
                ;       FRET
                ;       GETCH
                ;       GETHX
                ;       GETNM
                ;       HILO
                ;       NMOUT
                ;       PRVAL
                ;       REGDS
                ;       RGADR
                ;       RSTTF
                ;       SRET
                ;       STHF0
                ;       STHLF
                ;       VALDG
                ;       VALDL
                ;       -----
                ;
0000                    ORG     0H
                ;
                ;*************************************************************
                ;
                ;
                ;                       MONITOR EQUATES
                ;
                ;
                ;*************************************************************
                ;
                ;
001B            BRCHR EQU     1BH      ; CODE FOR BREAK CHARACTER (ESCAPE)
13FD            BRLOC EQU     13FDH    ; LOCATION OF USER BRANCH INSTRUCTION IN RAM
03FA            BRTAB EQU     3FAH     ; LOCATION OF START OF BRANCH TABLE IN ROM
0027            CMD   EQU     027H     ; COMMAND INSTRUCTION FOR USART INITIALIZATION
00FB            CNCTL EQU     0FBH     ; CONSOLE (USART) CONTROL PORT
00FA            CNIN  EQU     0FAH     ; CONSOLE INPUT PORT
00FA            CNOUT EQU     0FAH     ; CONSOLE OUTPUT PORT
00FB            CONST EQU     0FBH     ; CONSOLE STATUS INPUT PORT
000D            CR    EQU     0DH      ; CODE FOR CARRIAGE RETURN
1300            DATA  EQU     5*1024-256       ; START OF MONITOR RAM USAGE
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 3

  ØØ1B            ESC   EQU     1BH      ; CODE FOR ESCAPE CHARACTER
  ØØØF            HCHAR EQU     ØFH      ; MASK TO SELECT LOWER HEX CHAR FROM BYTE
  ØØFF            INVRT EQU     ØFFH     ; MASK TO INVERT HALF BYTE FLAG
  ØØØA            LF    EQU     ØAH      ; CODE FOR LINE FEED
  ØØØØ            LOWER EQU     Ø        ; DENOTES LOWER HALF OF BYTE IN ICMD
                  ;LSGNON       EQU     ---      ; LENGTH OF SIGNON MESSAGE - DEFINED LATE
  ØØCF            MODE  EQU     ØCFH     ; MODE SET FOR USART INITIALIZATION
                  ;MSTAK        EQU     ---      ; START OF MONITOR STACK - DEFINED LATER
                  ;NCMDS        EQU     ---      ; NUMBER OF VALID COMMANDS
  ØØØF            NEWLN EQU     ØFH      ; MASK FOR CHECKING MEMORY ADDR DISPLAY
  Ø07F            PRTYØ EQU     Ø7FH     ; MASK TO CLEAR PARITY BIT FROM CONSOLE CHAR
  13ED            REGS  EQU     DATA+255-18      ; START OF REGISTER SAVE AREA
  ØØØ2            RBR   EQU     2        ; MASK TO TEST RECEIVER STATUS
  ØØ38            RSTU  EQU     38H      ; TRANSFER LOCATION FOR RST 7 INSTRUCTION
                  ;RTABS        EQU     ---      ; SIZE OF ENTRY IN RTAB TABLE
  ØØ1B            TERM  EQU     1BH      ; CODE FOR ICMD TERMINATING CHARACTER (ESCAPE)
  ØØØ1            TRDY  EQU     1.       ; MASK TO TEST TRANSMITTER STATUS
  ØØFF            UPPER EQU     ØFFH     ; DENOTES UPPER HALF OF BYTE IN ICMD
                  ;
                  ;
                  ;*****************************************************************
                  ;
                  ;
                  ;                    MONITOR MACROS
                  ;
                  ;
                  ;*****************************************************************
                  ;
                  ;
      1           TRUE  MACRO   WHERE    ; BRANCH IF FUNCTION RETURNS TRUE (SUCCESS)
      1                 JC      WHERE
                        ENDM
                  ;
      1           FALSE MACRO   WHERE    ; BRANCH IF FUNCTION RETURNS FALSE (FAILURE)
      1                 JNC     WHERE
                        ENDM
                  ;
                  ;
                  ;*****************************************************************
                  ;
                  ;
                  ;                  USART INITIALIZATION CODE
                  ;
                  ;
                  ;*****************************************************************
                  ;
                  ;
                  ;   THE USART IS ASSUMED TO COME UP IN THE RESET POSITION (THIS
                  ;   FUNCTION IS TAKEN CARE OF BY THE HARDWARE). THE USART WILL
                  ;   BE INITIALIZED IN THE SAME WAY FOR EITHER A TTY OR CRT
                  ;   INTERFACE. THE FOLLOWING PARAMETERS ARE USED:
```

Mode:
2 Stopbits
Parity: Disable (odd)
8 Bits
64 X

```
8080 MACRO ASSEMBLER, VER 2.3  ERRORS = 0 PAGE 4

                        ;
                        ;       MODE INSTRUCTION
                        ;       ==== ===========
                        ;
                        ;       2 STOP BITS
                        ;       PARITY DISABLED
                        ;       8 BIT CHARACTERS
                        ;       BAUD RATE FACTOR OF 64
                        ;
                        ;       COMMAND INSTRUCTION
                        ;       ======= ===========
                        ;
                        ;       NO HUNT MODE
No Internal Reset       ;       NOT(RTS) FORCED TO 0
No Error Reset          ;       RECEIVE ENABLED
No Break                ;       DATA TERMINAL READY ≙ 0
                        ;       TRANSMIT ENABLED
                        ;
0000    3ECF                    MVI     A,MODE
0002    D3FB                    OUT     CNCTL   ; OUTPUT MODE SET TO USART
0004    3E27                    MVI     A,CMD
0006    D3FB                    OUT     CNCTL   ; OUTPUT COMMAND WORD TO USART
                        ;
                        ;*****************************************************************
                        ;
                        ;
                        ;                       RESTART ENTRY POINT
                        ;
                        ;
                        ;*****************************************************************
                        ;
                        ;
0008                    GO:
0008    22F313                  SHLD    LSAVE   ; SAVE HL REGISTERS
000B    E1                      POP     H       ; GET TOP OF STACK ENTRY
000C    22F513                  SHLD    PSAVE   ; ASSUME THIS IS LAST P COUNTER
000F    210000                  LXI     H,0     ; CLEAR HL
0012    39                      DAD     SP      ; GET STACK POINTER VALUE
0013    22F713                  SHLD    SSAVE   ; SAVE USER'S STACK POINTER
0016    21F313                  LXI     H,ASAVE+1       ; NEW VALUE FOR STACK POINTER
0019    F9                      SPHL            ; SET MONITOR STACK POINTER FOR REG SAVE
001A    F5                      PUSH    PSW     ; SAVE A AND FLAGS
001B    C5                      PUSH    B       ; SAVE B AND C
001C    D5                      PUSH    D       ; SAVE D AND E
                        ;
                        ;*****************************************************************
                        ;
                        ;
                        ;                       PRINT SIGNON MESSAGE
                        ;
```

C F
1 1 0 0 1 1 1 1 USART Mode

2 7
0 0 1 0 0 1 1 1 Command Instr.

```
8080 MACRO ASSEMBLER, VER 2.3  ERRORS = 0 PAGE 5

                    ;*************************************************************
                    ;
                    ;
  001D   219D03           LXI     H,SGNON ; GET ADDRESS OF SIGNON MESSAGE
  0020   060E             MVI     B,LSGNON        ; COUNTER FOR CHARACTERS IN MESSAGE
  0022              MSGL:
  0022   4E               MOV     C,M     ; FETCH NEXT CHAR TO C REG
  0023   CDE301           CALL    CO      ; SEND IT TO THE CONSOLE
  0026   23               INX     H       ; POINT TO NEXT CHARACTER
  0027   05               DCR     B       ; DECREMENT BYTE COUNTER
  0028   C22200           JNZ     MSGL    ; RETURN FOR NEXT CHARACTER
                    ;
                    ;
                    ;*****************************************************************
                    ;
                    ;
                    ;                    COMMAND RECOGNIZING ROUTINE
                    ;
                    ;
                    ;*****************************************************************
                    ;
                    ; FUNCTION: GETCM
                    ; INPUTS: NONE
                    ; OUTPUTS: NONE
                    ; CALLS: GETCH, ECHO, ERROR
                    ; DESTROYS: A,B,C,H,L,F/F'S
                    ; DESCRIPTION: GETCM RECEIVES AN INPUT CHARACTER FROM THE USER
                    ;               AND ATTEMPTS TO LOCATE THIS CHARACTER IN ITS COMMAND
                    ;               CHARACTER TABLE. IF SUCCESSFUL, THE ROUTINE
                    ;               CORRESPONDING TO THIS CHARACTER IS SELECTED FROM
                    ;               A TABLE OF COMMAND ROUTINE ADDRESSES, AND CONTROL
                    ;               IS TRANSFERRED TO THIS ROUTINE. IF THE CHARACTER
                    ;               DOES NOT MATCH ANY ENTRIES, CONTROL IS PASSED TO
                    ;               THE ERROR HANDLER.
                    ;
  002B        10    GETCM:
§ 002B   21ED13           LXI     H,MSTAK ; ALWAYS WANT TO RESET STACK PTR TO MONITOR
  002E   F9               SPHL            ; /STARTING VALUE SO ROUTINES NEEDN'T CLEAN UP
  002F   0E2E             MVI     C,'.'   ; PROMPT CHARACTER TO C
  0031   CDF401           CALL    ECHO    ; SEND PROMPT CHARACTER TO USER TERMINAL
  0034   C33B00           JMP     GTC03   ; WANT TO LEAVE ROOM FOR RST BRANCH
  0037   00         ;
  0038                    ORG     RSTU    ; ORG TO RST TRANSFER LOCATION
§ 0038   C3FD13           JMP     USRBR   ; JUMP TO USER BRANCH LOCATION
         CF0000     ;
  003B              GTC03:
  003B   CD1B02           CALL    GETCH   ; GET COMMAND CHARACTER TO A
  003E   CDF401           CALL    ECHO    ; ECHO CHARACTER TO USER
  0041   79               MOV     A,C     ; PUT COMMAND CHARACTER INTO ACCUMULATOR
  0042   010600           LXI     B,NCMDS ; C CONTAINS LOOP AND INDEX COUNT
  0045   21B903           LXI     H,CTAB  ; HL POINTS INTO COMMAND TABLE
```

```
8ØØØ MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 6

 ØØ48              GTCØ5:
 ØØ48   BE               CMP     M        ; COMPARE TABLE ENTRY AND CHARACTER
 ØØ49   CA54ØØ           JZ      GTC1Ø    ; BRANCH IF EQUAL - COMMAND RECOGNIZED
 ØØ4C   23               INX     H        ; ELSE, INCREMENT TABLE POINTER
 ØØ4D   ØD               DCR     C        ; DECREMENT LOOP COUNT
 ØØ4E   C248ØØ           JNZ     GTCØ5    ; BRANCH IF NOT AT TABLE END
 ØØ51   C3ØDØ2           JMP     ERROR    ; ELSE, COMMAND CHARACTER IS ILLEGAL
 ØØ54              GTC1Ø:
 ØØ54   21ABØ3           LXI     H,CADR   ; IF GOOD COMMAND, LOAD ADDRESS OF TABLE
                                          ; /OF COMMAND ROUTINE ADDRESSES
 ØØ57   Ø9               DAD     B        ; ADD WHAT IS LEFT OF LOOP COUNT
 ØØ58   Ø9               DAD     B        ; ADD AGAIN - EACH ENTRY IN CADR IS 2 BYTES LONG
 ØØ59   7E               MOV     A,M      ; GET LSP OF ADDRESS OF TABLE ENTRY TO A
 ØØ5A   23               INX     H        ; POINT TO NEXT BYTE IN TABLE
 ØØ5B   66               MOV     H,M      ; GET MSP OF ADDRESS OF TABLE ENTRY TO H
 ØØ5C   6F               MOV     L,A      ; PUT LSP OF ADDRESS OF TABLE ENTRY INTO L
 ØØ5D   E9               PCHL             ; NEXT INSTRUCTION COMES FROM COMMAND ROUTINE
                   ;
                   ;
                   ;*****************************************************************
                   ;
                   ;
                   ;              COMMAND IMPLEMENTING ROUTINES
                   ;
                   ;
                   ;*****************************************************************
                   ;
                   ;
                   ; FUNCTION: DCMD
                   ; INPUTS: NONE
                   ; OUTPUTS: NONE
                   ; CALLS: ECHO,NMOUT,HILO,GETCM,CROUT,GETNM
                   ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                   ; DESCRIPTION: DCMD IMPLEMENTS THE DISPLAY MEMORY (D) COMMAND
                   ;
 ØØ5E              DCMD:
 ØØ5E   ØEØ2             MVI     C,2      ; GET 2 NUMBERS FROM INPUT STREAM
 ØØ6Ø   CD57Ø2           CALL    GETNM
 ØØ63   D1               POP     D        ; ENDING ADDRESS TO DE
 ØØ64   E1               POP     H        ; STARTING ADDRESS TO HL
 ØØ65              DCMØ5:
 ØØ65   CDEEØ1           CALL    CROUT    ; ECHO CARRIAGE RETURN/LINE FEED
 ØØ68   7C               MOV     A,H      ; DISPLAY ADDRESS OF FIRST LOCATION IN LINE
 ØØ69   CDC3Ø2           CALL    NMOUT
 ØØ6C   7D               MOV     A,L      ; ADDRESS IS 2 BYTES LONG
 ØØ6D   CDC3Ø2           CALL    NMOUT
 ØØ7Ø              DCM1Ø:
 ØØ7Ø   ØE2Ø             MVI     C,''
 ØØ72   CDF4Ø1           CALL    ECHO     ; USE BLANK AS SEPARATOR
 ØØ75   7E               MOV     A,M      ; GET CONTENTS OF NEXT MEMORY LOCATION
 ØØ76   CDC3Ø2           CALL    NMOUT    ; DISPLAY CONTENTS
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 7

  ØØ79   CDBDØ1          CALL    BREAK   ; SEE IF USER WANTS OUT
       1          +      TRUE    DCM12   ; IF SO, BRANCH
  ØØ7C 1 DA85ØØ   +      JC      DCM12
  ØØ7F   CD9CØ2          CALL    HILO    ; SEE IF ADDRESS OF DISPLAYED LOCATION IS
                                         ; /GREATER THAN OR EQUAL TO ENDING ADDRESS
       1          +      FALSE   DCM15   ; IF NOT, MORE TO DISPLAY
  ØØ82 1 D28BØØ   +      JNC     DCM15
  ØØ85            DCM12:
  ØØ85   CDEEØ1          CALL    CROUT   ; CARRIAGE RETURN/LINE FEED TO END LINE
  ØØ88   C32BØØ          JMP     GETCM   ; ALL DONE
  ØØ8B            DCM15:
  ØØ8B   23              INX     H       ; IF MORE TO GO, POINT TO NEXT LOC TO DISPLAY
  ØØ8C   7D              MOV     A,L     ; GET LOW ORDER BITS OF NEW ADDRESS
  ØØ8D   E6ØF            ANI     NEWLN   ; SEE IF LAST HEX DIGIT OF ADDRESS DENOTES
                                         ; /START OF NEW LINE
  ØØ8F   C27ØØØ          JNZ     DCM1Ø   ; NO - NOT AT END OF LINE
  ØØ92   C365ØØ          JMP     DCMØ5   ; YES - START NEW LINE WITH ADDRESS
                 ;
                 ;
                 ;*****************************************************************
                 ;
                 ;
                 ; FUNCTION: GCMD
                 ; INPUTS: NONE
                 ; OUTPUTS: NONE
                 ; CALLS: ERROR,GETHX,RSTTF
                 ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                 ; DESCRIPTION: GCMD IMPLEMENTS THE BEGIN EXECUTION (G) COMMAND.
                 ;
  ØØ95           GCMD:
  ØØ95   CD22Ø2          CALL    GETHX   ; GET ADDRESS (IF PRESENT) FROM INPUT STREAM
       1          +      FALSE   GCMØ5   ; BRANCH IF NO NUMBER PRESENT
  ØØ98 1 D2AAØØ   +      JNC     GCMØ5
  ØØ9B   7A              MOV     A,D     ; ELSE, GET TERMINATOR
  ØØ9C   FEØD            CPI     CR      ; SEE IF CARRIAGE RETURN
  ØØ9E   C2ØDØ2          JNZ     ERROR   ; ERROR IF NOT PROPERLY TERMINATED
  ØØA1   21F513          LXI     H,PSAVE ; WANT NUMBER TO REPLACE SAVE PGM COUNTER
  ØØA4   71  10          MOV     M,C
  ØØA5   23              INX     H
  ØØA6   7Ø              MOV     M,B
  ØØA7   C3BØØØ          JMP     GCM1Ø
  ØØAA           GCMØ5:
  ØØAA   7A              MOV     A,D     ; IF NO STARTING ADDRESS, MAKE SURE THAT
  ØØAB   FEØD            CPI     CR      ; /CARRIAGE RETURN TERMINATED COMMAND
  ØØAD   C2ØDØ2          JNZ     ERROR   ; ERROR IF NOT
  ØØBØ           GCM1Ø:
  ØØBØ   C32EØ3          JMP     RSTTF   ; RESTORE REGISTERS AND BEGIN EXECUTION

                 ;
                 ;
                 ;*****************************************************************
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 8

                        ;
                        ;
                        ; FUNCTION: ICMD
                        ; INPUTS: NONE
                        ; OUTPUTS: NONE
                        ; CALLS: ERROR,ECHO,GETCH,VALDL,VALDG,CNVBN,STHLF,GETNM,CROUT
                        ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                        ; DESCRIPTION: ICMD IMPLEMENTS THE INSERT CODE INTO MEMORY (I) COMMAND.
                        ;
  ØØB3                  ICMD:
  ØØB3   ØEØ1                 MVI     C,1
  ØØB5   CD57Ø2               CALL    GETNM   ; GET SINGLE NUMBER FROM INPUT STREAM
  ØØB8   3EFF                 MVI     A,UPPER
  ØØBA   32F913               STA     TEMP    ; TEMP WILL HOLD THE UPPER/LOWER HALF BYTE FLAG
  ØØBD   D1                   POP     D       ; ADDRESS OF START TO DE
  ØØBE                  ICMØ5:
  ØØBE   CD1BØ2               CALL    GETCH   ; GET A CHARACTER FROM INPUT STREAM
  ØØC1   4F                   MOV     C,A
  ØØC2   CDF4Ø1               CALL    ECHO    ; ECHO IT
  ØØC5   79                   MOV     A,C     ; PUT CHARACTER BACK INTO A
  ØØC6   FE1B                 CPI     TERM    ; SEE IF CHARACTER IS A TERMINATING CHARACTER
  ØØC8   CAF4ØØ               JZ      ICM25   ; IF SO, ALL DONE ENTERING CHARACTERS
  ØØCB   CD8AØ3               CALL    VALDL   ; ELSE, SEE IF VALID DELIMITER
       1                +      TRUE    ICMØ5   ; IF SO SIMPLY IGNORE THIS CHARACTER
  ØØCE 1 DABEØØ         +      JC      ICMØ5
  ØØD1   CD6FØ3               CALL    VALDG   ; ELSE, CHECK TO SEE IF VALID HEX DIGIT
       1                +      FALSE   ICM2Ø   ; IF NOT, BRANCH TO HANDLE ERROR CONDITION
  ØØD4 1 D2EEØØ         +      JNC     ICM2Ø
  ØØD7   CDDAØ1               CALL    CNVBN   ; CONVERT DIGIT TO BINARY
  ØØDA   4F                   MOV     C,A     ; MOVE RESULT TO C
  ØØDB   CD5ØØ3               CALL    STHLF   ; STORE IN APPROPRIATE HALF WORD
  ØØDE   3AF913               LDA     TEMP    ; GET HALF BYTE FLAG
  ØØE1   B7                   ORA     A       ; SET F/F'S
  ØØE2   C2E6ØØ               JNZ     ICM1Ø   ; BRANCH IF FLAG SET FOR UPPER
  ØØE5   13                   INX     D       ; IF LOWER, INC ADDRESS OF BYTE TO STORE IN
  ØØE6                  ICM1Ø:
  ØØE6   EEFF                 XRI     INVRT   ; TOGGLE STATE OF FLAG
  ØØE8   32F913               STA     TEMP    ; PUT NEW VALUE OF FLAG BACK
  ØØEB   C3BEØØ               JMP     ICMØ5   ; PROCESS NEXT DIGIT
  ØØEE                  ICM2Ø:
  ØØEE   CD45Ø3               CALL    STHFØ   ; ILLEGAL CHARACTER
  ØØF1   C3ØDØ2               JMP     ERROR   ; MAKE SURE ENTIRE BYTE FILLED THEN ERROR
  ØØF4                  ICM25:
  ØØF4   CD45Ø3               CALL    STHFØ   ; HERE FOR ESCAPE CHARACTER - INPUT IS DONE
  ØØF7   CDEEØ1               CALL    CROUT   ; ADD CARRIAGE RETURN
  ØØFA   C32BØØ               JMP     GETCM
                        ;
                        ;
                        ;*****************************************************************
                        ;
                        ;
```

```
8080 MACRO ASSEMBLER, VER 2.3  ERRORS = 0 PAGE 9

                    ; FUNCTION: MCMD
                    ; INPUTS: NONE
                    ; OUTPUTS: NONE
                    ; CALLS: GETCM,HILO,GETNM
                    ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                    ; DESCRIPTION: MCMD IMPLEMENTS THE MOVE DATA IN MEMORY (M) COMMAND.
                    ;
00FD                MCMD:
00FD   0E03               MVI     C,3
00FF   CD5702             CALL    GETNM    ; GET 3 NUMBERS FROM INPUT STREAM
0102   C1                 POP     B        ; DESTINATION ADDRESS TO BC
0103   E1                 POP     H        ; ENDING ADDRESS TO HL
0104   D1                 POP     D        ; STARTING ADDRESS TO DE
0105                MCM05:
0105   E5                 PUSH    H        ; SAVE ENDING ADDRESS
0106   62                 MOV     H,D
0107   6B                 MOV     L,E      ; SOURCE ADDRESS TO HL
0108   7E                 MOV     A,M      ; GET SOURCE BYTE
0109   60                 MOV     H,B
010A   69                 MOV     L,C      ; DESTINATION ADDRESS TO HL
010B   77                 MOV     M,A      ; MOVE BYTE TO DESTINATION
010C   03                 INX     B        ; INCREMENT DESTINATION ADDRESS
010D   78                 MOV     A,B
010E   B1                 ORA     C        ; TEST FOR DESTINATION ADDRESS OVERFLOW
010F   CA2B00             JZ      GETCM    ; IF SO, CAN TERMINATE COMMAND
0112   13                 INX     D        ; INCREMENT SOURCE ADDRESS
0113   E1                 POP     H        ; ELSE, GET BACK ENDING ADDRESS
0114   CD9C02             CALL    HILO     ; SEE IF ENDING ADDR>=SOURCE ADDR
     1               +    FALSE   GETCM    ; IF NOT, COMMAND IS DONE
0117 1 D22B00        +    JNC     GETCM
011A   C3050 1            JMP     MCM05    ; MOVE ANOTHER BYTE
                    ;
                    ;
                    ;*****************************************************************
                    ;
                    ;
                    ; FUNCTION: SCMD
                    ; INPUTS: NONE
                    ; OUTPUTS: NONE
                    ; CALLS: GETHX,GETCM,NMOUT,ECHO
                    ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                    ; DESCRIPTION: SCMD IMPLEMENTS THE SUBSTITUTE INTO MEMORY (S) COMMAND.
                    ;
011D                SCMD:
011D   CD2202             CALL    GETHX    ; GET A NUMBER, IF PRESENT, FROM INPUT
0120   C5                 PUSH    B
0121   E1                 POP     H        ; GET NUMBER TO HL - DENOTES MEMORY LOCATION
0122                SCM05:
0122   7A                 MOV     A,D      ; GET TERMINATOR
0123   FE20               CPI     ' '      ; SEE IF SPACE
0125   CA2D01             JZ      SCM10    ; YES - CONTINUE PROCESSING
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 1Ø

  Ø128   FE2C          CPI     ',',      ; ELSE, SEE IF COMMA
  Ø12A   C22BØØ        JNZ     GETCM     ; NO - TERMINATE COMMAND
  Ø12D             SCM1Ø:
  Ø12D   7E            MOV     A,M       ; GET CONTENTS OF SPECIFIED LOCATION TO A
  Ø12E   CDC3Ø2        CALL    NMOUT     ; DISPLAY CONTENTS ON CONSOLE
  Ø131   ØE2D          MVI     C,'-'
  Ø133   CDF4Ø1        CALL    ECHO      ; USE DASH FOR SEPARATOR
  Ø136   CD22Ø2        CALL    GETHX     ; GET NEW VALUE FOR MEMORY LOCATION, IF ANY
       1           +   FALSE   SCM15     ; IF NO VALUE PRESENT, BRANCH
  Ø139 1 D23DØ1    +   JNC     SCM15
  Ø13C   71            MOV     M,C       ; ELSE, STORE LOWER 8 BITS OF NUMBER ENTERED
  Ø13D             SCM15:
  Ø13D   23            INX     H         ; INCREMENT ADDRESS OF MEMORY LOCATION TO VIEW
  Ø13E   C322Ø1        JMP     SCMØ5
                   ;
                   ;
                   ;*****************************************************************
                   ;
                   ;
                   ; FUNCTION: XCMD
                   ; INPUTS: NONE
                   ; OUTPUTS: NONE
                   ; CALLS: GETCH,ECHO,REGDS,GETCM,ERROR,RGADR,NMOUT,CROUT,GETHX
                   ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                   ; DESCRIPTION: XCMD IMPLEMENTS THE REGISTER EXAMINE AND CHANGE (X)
                   ;               COMMAND.
                   ;
  Ø141             XCMD:
  Ø141   CD1BØ2        CALL    GETCH     ; GET REGISTER IDENTIFIER
  Ø144   4F            MOV     C,A
  Ø145   CDF4Ø1        CALL    ECHO      ; ECHO IT
  Ø148   79            MOV     A,C
  Ø149   FEØD          CPI     CR
  Ø14B   C254Ø1        JNZ     XCMØ5     ; BRANCH IF NOT CARRIAGE RETURN
  Ø14E   CDE6Ø2        CALL    REGDS     ; ELSE, DISPLAY REGISTER CONTENTS
  Ø151   C32BØØ        JMP     GETCM     ; THEN TERMINATE COMMAND
  Ø154             XCMØ5:
  Ø154   4F            MOV     C,A       ; GET REGISTER IDENTIFIER TO C
  Ø155   CD17Ø3        CALL    RGADR     ; CONVERT IDENTIFIER INTO RTAB TABLE ADDR
  Ø158   C5            PUSH    B
  Ø159   E1            POP     H         ; PUT POINTER TO REGISTER ENTRY INTO HL
  Ø15A   ØE2Ø          MVI     C,' '
  Ø15C   CDF4Ø1        CALL    ECHO      ; ECHO SPACE TO USER
  Ø15F   79            MOV     A,C
  Ø16Ø   32F913        STA     TEMP      ; PUT SPACE INTO TEMP AS DELIMITER
  Ø163             XCM1Ø:
  Ø163   3AF913        LDA     TEMP      ; GET TERMINATOR
  Ø166   FE2Ø          CPI     ' '       ; SEE IF A BLANK
  Ø168   CA7ØØ1        JZ      XCM15     ; YES - GO CHECK POINTER INTO TABLE
  Ø16B   FE2C          CPI     ','       ; NO - SEE IF COMMA
  Ø16D   C22BØØ        JNZ     GETCM     ; NO - MUST BE CARRIAGE RETURN TO END COMMAND
```

```
8080 MACRO ASSEMBLER, VER 2.3  ERRORS = 0 PAGE 11

  0170              XCM15:
  0170   7E               MOV     A,M
  0171   B7               ORA     A        ; SET F/F'S
  0172   C27B01           JNZ     XCM18    ; BRANCH IF NOT AT END OF TABLE
  0175   CDEE01           CALL    CROUT    ; ELSE, OUTPUT CARRIAGE RETURN LINE FEED
  0178   C32B00           JMP     GETCM    ; AND EXIT
  017B              XCM18:
  017B   E5               PUSH    H        ; PUT POINTER ON STACK
  017C   5E               MOV     E,M
  017D   1613             MVI     D,DATA SHR 8    ; FETCH ADDRESS OF SAVE LOCATION FROM TABI
  017F   23               INX     H
  0180   46               MOV     B,M      ; FETCH LENGTH FLAG FROM TABLE
  0181   D5               PUSH    D        ; SAVE ADDRESS OF SAVE LOCATION
  0182   D5               PUSH    D
  0183   E1               POP     H        ; MOVE ADDRESS TO HL
  0184   C5               PUSH    B        ; SAVE LENGTH FLAG
  0185   7E               MOV     A,M      ; GET 8 BITS OF REGISTER FROM SAVE LOCATION
  0186   CDC302           CALL    NMOUT    ; DISPLAY IT
  0189   F1               POP     PSW      ; GET BACK LENGTH FLAG
  018A   F5               PUSH    PSW      ; SAVE IT AGAIN
  018B   B7               ORA     A        ; SET F/F'S
  018C   CA9401           JZ      XCM20    ; IF 8 BIT REGISTER, NOTHING MORE TO DISPLAY
  018F   2B               DCX     H        ; ELSE, FOR 16 BIT REGISTER, GET LOWER 8 BITS
  0190   7E               MOV     A,M
  0191   CDC302           CALL    NMOUT    ; DISPLAY THEM
  0194              XCM20:
  0194   0E2D             MVI     C,'-'
  0196   CDF401           CALL    ECHO     ; USE DASH AS SEPARATOR
  0199   CD2202           CALL    GETHX    ; SEE IF THERE IS A VALUE TO PUT INTO REGISTER
       1            +     FALSE   XCM30    ; NO - GO CHECK FOR NEXT REGISTER
  019C 1 D2B401     +     JNC     XCM30
  019F   7A               MOV     A,D
  01A0   32F913           STA     TEMP     ; ELSE, SAVE THE TERMINATOR FOR NOW
  01A3   F1               POP     PSW      ; GET BACK LENGTH FLAG
  01A4   E1               POP     H        ; PUT ADDRESS OF SAVE LOCATION INTO HL
  01A5   B7               ORA     A        ; SET F/F'S
  01A6   CAAB01           JZ      XCM25    ; IF 8 BIT REGISTER, BRANCH
  01A9   70               MOV     M,B      ; SAVE UPPER 8 BITS
  01AA   2B               DCX     H        ; POINT TO SAVE LOCATION FOR LOWER 8 BITS
  01AB              XCM25:
  01AB   71               MOV     M,C      ; STORE ALL OF 8 BIT OR LOWER 1/2 OF 16 BIT REG
  01AC              XCM27:
  01AC   110300           LXI     D,RTABS  ; SIZE OF ENTRY IN RTAB TABLE
  01AF   E1               POP     H        ; POINTER INTO REGISTER TABLE RTAB
  01B0   19               DAD     D        ; ADD ENTRY SIZE TO POINTER
  01B1   C36301           JMP     XCM10    ; DO NEXT REGISTER
  01B4              XCM30:
  01B4   7A               MOV     A,D      ; GET TERMINATOR
  01B5   32F913           STA     TEMP     ; SAVE IN MEMORY
  01B8   D1               POP     D        ; CLEAR STACK OF LENGTH FLAG AND ADDRESS
  01B9   D1               POP     D        ; /OF SAVE LOCATION
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 12

  Ø1BA   C3ACØ1          JMP     XCM27    ; GO INCREMENT REGISTER TABLE POINTER
                    ;
                    ;
                    ;*****************************************************************
                    ;
                    ;
                    ;                    UTILITY ROUTINES
                    ;
                    ;
                    ;*****************************************************************
                    ;
                    ;
                    ; FUNCTION: BREAK
                    ; INPUTS: NONE
CTER INPUT          ; OUTPUTS: CARRY - 1 IF ESCAPE CHARA
                    ;                - Ø IF ANY OTHER CHARACTER OR NO CHARACTER PENDING
                    ; CALLS: NOTHING
                    ; DESTROYS: A,F/F'S
                    ; DESCRIPTION: BREAK IS USED TO SENSE AN ESCAPE CHARACTER FROM
                    ;               THE USER. IF NO CHARACTER IS PENDING, OR IF THE
                    ;               PENDING CHARACTER IS NOT THE ESCAPE, THEN A FAILURE
                    ;               RETURN (CARRY=Ø) IS TAKEN. IN THIS CASE,
                                                                              THE
                    ;               PENDING CHARACTER (IF ANY) IS LOST. IF THE PENDING
                    ;               CHARACTER IS AN ESCAPE CHARACTER, BREAK TAKES A SUCCESS
                    ;               RETURN (CARRY=1).
                    ;
  Ø1BD              BREAK:
  Ø1BD   DBFB            IN      CONST    ; GET CONSOLE STATUS
  Ø1BF   E6Ø2            ANI     RBR      ; SEE IF CHARACTER PENDING
  Ø1C1   CA18Ø2          JZ      FRET     ; NO - TAKE FAILURE RETURN
  Ø1C4   DBFA            IN      CNIN     ; YES - PICK UP CHARACTER
  Ø1C6   E67F            ANI     PRTYØ    ; STRIP OFF PARITY BIT
  Ø1C8   FE1B            CPI     BRCHR    ; SEE IF BREAK CHARACTER
  Ø1CA   CA43Ø3          JZ      SRET     ; YES - SUCCESS RETURN
  Ø1CD   C318Ø2          JMP     FRET     ; NO - FAILURE RETURN - CHARACTER LOST
                    ;
                    ;
                    ;*****************************************************************
                    ;
                    ;
                    ; FUNCTION: CI
                    ; INPUTS: NONE
                    ; OUTPUTS: A - CHARACTER FROM CONSOLE
                    ; CALLS: NOTHING
                    ; DESTROYS: A,F/F'S
                    ; DESCRIPTION: CI WAITS UNTIL A CHARACTER HAS BEEN ENTERED AT THE
                    ;               CONSOLE AND THEN RETURNS THE CHARACTER, VIA THE A
                    ;               REGISTER, TO THE CALLING ROUTINE. THIS ROUTINE
                    ;               IS CALLED BY THE USER VIA A JUMP TABLE IN RAM.
                    ;
  Ø1DØ              CI:
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 13

  Ø1DØ   DBFB          IN      CONST    ; GET STATUS OF CONSOLE
  Ø1D2   E6Ø2          ANI     RBR      ; CHECK FOR RECEIVER BUFFER READY
  Ø1D4   CADØØ1        JZ      CI       ; NOT YET - WAIT
  Ø1D7   DBFA          IN      CNIN     ; READY SO GET CHARACTER
  Ø1D9   C9            RET
                 ;
                 ;
                 ;*****************************************************************
                 ;
                 ;
                 ; FUNCTION: CNVBN
                 ; INPUTS: C - ASCII CHARACTER 'Ø'-'9' OR  'A'-'F'
                 ; OUTPUTS: A - Ø TO F HEX
                 ; CALLS: NOTHING
                 ; DESTROYS: A,F/F'S
                 ; DESCRIPTION: CNVBN CONVERTS THE ASCII REPRESENTATION OF A HEX
                 ;               CNVBN INTO ITS CORRESPONDING BINARY VALUE. CNVBN
                 ;               DOES NOT CHECK THE VALIDITY OF ITS INPUT.
                 ;
  Ø1DA           CNVBN:
  Ø1DA   79            MOV     A,C
  Ø1DB   D63Ø          SUI     'Ø'      ; SUBTRACT CODE FOR 'Ø' FROM ARGUMENT
  Ø1DD   FEØA          CPI     1Ø       ; WANT TO TEST FOR RESULT OF Ø TO 9
  Ø1DF   F8            RM               ; IF SO, THEN ALL DONE
  Ø1EØ   D6Ø7          SUI     7        ; ELSE, RESULT BETWEEN 17 AND 23 DECIMAL
  Ø1E2   C9            RET              ; SO RETURN AFTER SUBTRACTING BIAS OF 7
                 ;
                 ;
                 ;*****************************************************************
                 ;
                 ;
                 ; FUNCTION: CO
                 ; INPUTS: C - CHARACTER TO OUTPUT TO CONSOLE
                 ; OUTPUTS: C - CHARACTER OUTPUT TO CONSOLE
                 ; CALLS: NOTHING
                 ; DESTROYS: A,F/F'S
                 ; DESCRIPTION: CO WAITS UNTIL THE CONSOLE IS READY TO ACCEPT A CHARACTE
                 ;               AND THEN SENDS THE INPUT ARGUMENT TO THE CONSOLE.
                 ;
  Ø1E3           CO:
  Ø1E3   DBFB          IN      CONST    ; GET STATUS OF CONSOLE
  Ø1E5   E6Ø1          ANI     TRDY     ; SEE IF TRANSMITTER READY
  Ø1E7   CAE3Ø1        JZ      CO       ; NO - WAIT
  Ø1EA   79            MOV     A,C      ; ELSE, MOVE CHARACTER TO A REGISTER FOR OUTPUT
  Ø1EB   D3FA          OUT     CNOUT    ; SEND TO CONSOLE
  Ø1ED   C9            RET
                 ;
                 ;
                 ;*****************************************************************
                 ;
                 ;
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 14

                    ; FUNCTION CROUT
                    ; INPUTS: NONE
                    ; OUTPUTS: NONE
                    ; CALLS: ECHO
                    ; DESTROYS: A,B,C,F/F'S
                    ; DESCRIPTION: CROUT SENDS A CARRIAGE RETURN (AND HENCE A LINE
                    ;               FEED) TO THE CONSOLE.
                    ;
  Ø1EE              CROUT:
  Ø1EE   ØEØD            MVI     C,CR
  Ø1FØ   CDF4Ø1          CALL    ECHO
  Ø1F3   C9              RET
                    ;
                    ;
                    ;*******************************************************************
                    ;
                    ;
                    ; FUNCTION: ECHO
                    ; INPUTS: C - CHARACTER TO ECHO TO TERMINAL
                    ; OUTPUTS: C - CHARACTER ECHOED TO TERMINAL
                    ; CALLS: CO
                    ; DESTROYS: A,B,F/F'S
                    ; DESCRIPTION: ECHO TAKES A SINGLE CHARACTER AS INPUT AND, VIA
                    ;               THE MONITOR, SENDS THAT CHARACTER TO THE USER
                    ;               TERMINAL. A CARRIAGE RETURN IS ECHOED AS A CARRIAGE
                    ;               RETURN LINE FEED, AND AN ESCAPE CHARACTER IS ECHOED AS $
                    ;
  Ø1F4              ECHO:
  Ø1F4   41              MOV     B,C      ; SAVE ARGUMENT
  Ø1F5   3E1B            MVI     A,ESC
  Ø1F7   B8              CMP     B        ; SEE IF ECHOING AN ESCAPE CHARACTER
  Ø1F8   C2FDØ1          JNZ     ECHØ5    ; NO - BRANCH
  Ø1FB   ØE24            MVI     C, '$'   ; YES - ECHO AS $
  Ø1FD              ECHØ5:
  Ø1FD   CDE3Ø1          CALL    CO       ; DO OUTPUT THROUGH MONITOR
  Ø2ØØ   3EØD            MVI     A,CR
  Ø2Ø2   B8              CMP     B        ; SEE IF CHARACTER ECHOED WAS A CARRIAGE RETURN
  Ø2Ø3   C2ØBØ2          JNZ     ECH1Ø    ; NO - NO NEED TO TAKE SPECIAL ACTION
  Ø2Ø6   ØEØA            MVI     C,LF     ; YES - WANT TO ECHO LINE FEED, TOO
  Ø2Ø8   CDE3Ø1          CALL    CO
  Ø2ØB              ECH1Ø:
  Ø2ØB   48              MOV     C,B      ; RESTORE ARGUMENT
  Ø2ØC   C9              RET
                    ;
                    ;
                    ;*******************************************************************
                    ;
                    ;
                    ; FUNCTION: ERROR
                    ; INPUTS: NONE
                    ; OUTPUTS: NONE
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 15

                    ; CALLS: ECHO,CROUT,GETCM
                    ; DESTROYS: A,B,C,F/F'S
                    ; DESCRIPTION: ERROR PRINTS THE ERROR CHARACTER (CURRENTLY AN ASTERISK)
                    ;              ON THE CONSOLE, FOLLOWED BY A CARRIAGE RETURN-LINE FEED,
                    ;              AND THEN RETURNS CONTROL TO THE COMMAND RECOGNIZER.
                    ;
 Ø2ØD               ERROR:
 Ø2ØD   ØE2A              MVI     C,'*'
 Ø2ØF   CDF4Ø1            CALL    ECHO     ; SEND * TO CONSOLE
 Ø212   CDEEØ1            CALL    CROUT    ; SKIP TO BEGINNING OF NEXT LINE
 Ø215   C32BØØ            JMP     GETCM    ; TRY AGAIN FOR ANOTHER COMMAND
                    ;
                    ;
                    ;*********************************************************************
                    ;
                    ;
                    ; FUNCTION: FRET
                    ; INPUTS: NONE
                    ; OUTPUTS: CARRY - ALWAYS Ø
                    ; CALLS: NOTHING
                    ; DESTROYS: CARRY
                    ; DESCRIPTION: FRET IS JUMPED TO BY ANY ROUTINE THAT WISHES TO
                    ;               INDICATE FAILURE ON RETURN. FRET SETS THE CARRY
                    ;               FALSE, DENOTING FAILURE, AND THEN RETURNS TO THE
                    ;               CALLER OF THE ROUTINE INVOKING FRET.
                    ;
 Ø218               FRET:
 Ø218   37                STC              ; FIRST SET CARRY TRUE
 Ø219   3F                CMC              ; THEN COMPLEMENT IT TO MAKE IT FALSE
 Ø21A   C9                RET              ; RETURN APPROPRIATELY
                    ;
                    ;
                    ;*********************************************************************
                    ;
                    ;
                    ; FUNCTION: GETCH
                    ; INPUTS: NONE
                    ; OUTPUTS: C - NEXT CHARACTER IN INPUT STREAM
                    ; CALLS: CI
                    ; DESTROYS: A,C,F/F'S
                    ; DESCRIPTION: GETCH RETURNS THE NEXT CHARACTER IN THE INPUT STREAM
                    ;               TO THE CALLING PROGRAM.
                    ;
 Ø21B               GETCH:
 Ø21B   CDDØØ1            CALL    CI       ; GET CHARACTER FROM TERMINAL
 Ø21E   E67F              ANI     PRTYØ    ; TURN OFF PARITY BIT IN CASE SET BY CONSOLE
 Ø22Ø   4F                MOV     C,A      ; PUT VALUE IN C REGISTER FOR RETURN
 Ø221   C9                RET
                    ;
                    ;
                    ;*********************************************************************
```

```
8080 MACRO ASSEMBLER, VER 2.3  ERRORS = 0 PAGE 16

                    ;
                    ;
                    ; FUNCTION: GETHX
                    ; INPUTS: NONE
                    ; OUTPUTS: BC - 16 BIT INTEGER
                    ;          D - CHARACTER WHICH TERMINATED THE INTEGER
                    ;          CARRY - 1 IF FIRST CHARACTER NOT DELIMITER
                    ;                - 0 IF FIRST CHARACTER IS DELIMITER
                    ; CALLS: GETCH,ECHO,VALDL,VALDG,CNVBN,ERROR
                    ; DESTROYS: A,B,C,D,E,F/F'S
                    ; DESCRIPTION: GETHX ACCEPTS A STRING OF HEX DIGITS FROM THE INPUT
                    ;               STREAM AND RETURNS THEIR VALUE AS A 16 BIT BINARY
                    ;               INTEGER. IF MORE THAN 4 HEX DIGITS ARE ENTERED,
                    ;               ONLY THE LAST 4 ARE USED. THE NUMBER TERMINATES WHEN
                    ;               A VALID DELIMITER IS ENCOUNTERED. THE DELIMITER IS
                    ;               ALSO RETURNED AS AN OUTPUT OF THE FUNCTION. ILLEGAL
                    ;               CHARACTERS (NOT HEX DIGITS OR DELIMITERS) CAUSE AN
                    ;               ERROR INDICATION. IF THE FIRST (VALID) CHARACTER
                    ;               ENCOUNTERED IN THE INPUT STREAM IS NOT A DELIMITER,
                    ;               GETHX WILL RETURN WITH THE CARRY BIT SET TO 1;
                    ;               OTHERWISE, THE CARRY BIT IS SET TO 0 AND THE CONTENTS
                    ;               OF BC ARE UNDEFINED.
                    ;
  0222              GETHX:
  0222   E5               PUSH    H        ; SAVE HL
  0223   210000           LXI     H,0      ; INITIALIZE RESULT
  0226   1E00             MVI     E,0      ; INITIALIZE DIGIT FLAG TO FALSE
  0228              GHX05:
  0228   CD1B02           CALL    GETCH    ; GET A CHARACTER
  022B   4F               MOV     C,A
  022C   CDF401           CALL    ECHO     ; ECHO THE CHARACTER
  022F   CD8A03           CALL    VALDL    ; SEE IF DELIMITER
       1            +     FALSE   GHX10    ; NO - BRANCH
  0232 1 D24102     +     JNC     GHX10
  0235   51               MOV     D,C      ; YES - ALL DONE, BUT WANT TO RETURN DELIMITER
  0236   E5               PUSH    H
  0237   C1               POP     B        ; MOVE RESULT TO BC
  0238   E1               POP     H        ; RESTORE HL
  0239   7B               MOV     A,E      ; GET FLAG
  023A   B7               ORA     A        ; SET F/F'S
  023B   C24303           JNZ     SRET     ; IF FLAG NON-0, A NUMBER HAS BEEN FOUND
  023E   CA1802           JZ      FRET     ; ELSE, DELIMITER WAS FIRST CHARACTER
  0241              GHX10:
  0241   CD6F03           CALL    VALDG    ; IF NOT DELIMITER, SEE IF DIGIT
       1            +     FALSE   ERROR    ; ERROR IF NOT A VALID DIGIT, EITHER
  0244 1 D20D02     +     JNC     ERROR
  0247   CDDA01           CALL    CNVBN    ; CONVERT DIGIT TO ITS BINARY VALUE
  024A   1EFF             MVI     E,0FFH   ; SET DIGIT FLAG NON-0
  024C   29               DAD     H        ; *2
  024D   29               DAD     H        ; *4
  024E   29               DAD     H        ; *8
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 17

  Ø24F   29              DAD     H        ; *16
  Ø25Ø   Ø6ØØ            MVI     B,Ø      ; CLEAR UPPER 8 BITS OF BC PAIR
  Ø252   4F              MOV     C,A      ; BINARY VALUE OF CHARACTER INTO C
  Ø253   Ø9              DAD     B        ; ADD THIS VALUE TO PARTIAL RESULT
  Ø254   C328Ø2          JMP     GHXØ5    ; GET NEXT CHARACTER
                   ;
                   ;
                   ;*****************************************************************
                   ;
                   ;
                   ; FUNCTION: GETNM
                   ; INPUTS: C - COUNT OF NUMBERS TO FIND IN INPUT STREAM
                   ; OUTPUTS: TOP OF STACK - NUMBERS FOUND IN REVERSE ORDER (LAST ON TOP
                   ;                          OF STACK)
                   ; CALLS: GETHX,HILO,ERROR
                   ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                   ; DESCRIPTION: GETNM FINDS A SPECIFIED COUNT OF NUMBERS, BETWEEN 1
                   ;               AND 3, INCLUSIVE,  IN THE INPUT
                   ;               STREAM AND RETURNS THEIR VALUES ON THE STACK. IF 2
                   ;               OR MORE NUMBERS ARE REQUESTED, THEN THE FIRST MUST BE
                   ;               LESS THAN OR EQUAL TO THE SECOND, OR THE FIRST AND
                   ;               SECOND NUMBERS WILL BE SET EQUAL. THE LAST NUMBER
                   ;               REQUESTED MUST BE TERMINATED BY A CARRIAGE RETURN
                   ;               OR AN ERROR INDICATION WILL RESULT.
                   ;
  Ø257             GETNM:
  Ø257   2EØ3            MVI     L,3      ; PUT MAXIMUM ARGUMENT COUNT INTO L
  Ø259   79              MOV     A,C      ; GET THE ACTUAL ARGUMENT COUNT
  Ø25A   E6Ø3            ANI     3        ; FORCE TO MAXIMUM OF 3
  Ø25C   C8              RZ               ; IF Ø, DON'T BOTHER TO DO ANYTHING
  Ø25D   67              MOV     H,A      ; ELSE, PUT ACTUAL COUNT INTO H
  Ø25E             GNMØ5:
  Ø25E   CD22Ø2          CALL    GETHX    ; GET A NUMBER FROM INPUT STREAM
       1         +       FALSE   ERROR    ; ERROR IF NOT THERE - TOO FEW NUMBERS
  Ø261 1 D2ØDØ2    +     JNC     ERROR
  Ø264   C5              PUSH    B        ; ELSE, SAVE NUMBER ON STACK
  Ø265   2D              DCR     L        ; DECREMENT MAXIMUM ARGUMENT COUNT
  Ø266   25              DCR     H        ; DECREMENT ACTUAL ARGUMENT COUNT
  Ø267   CA73Ø2          JZ      GNM1Ø    ; BRANCH IF NO MORE NUMBERS WANTED
  Ø26A   7A              MOV     A,D      ; ELSE, GET NUMBER TERMINATOR TO A
  Ø26B   FEØD            CPI     CR       ; SEE IF CARRIAGE RETURN
  Ø26D   CAØDØ2          JZ      ERROR    ; ERROR IF SO - TOO FEW NUMBERS
  Ø27Ø   C35EØ2          JMP     GNMØ5    ; ELSE, PROCESS NEXT NUMBER
  Ø273             GNM1Ø:
  Ø273   7A              MOV     A,D      ; WHEN COUNT Ø, CHECK LAST TERMINATOR
  Ø274   FEØD            CPI     CR
  Ø276   C2ØDØ2          JNZ     ERROR    ; ERROR IF NOT CARRIAGE RETURN
  Ø279   Ø1FFFF          LXI     B,ØFFFFH         ; HL GETS LARGEST NUMBER
  Ø27C   7D              MOV     A,L      ; GET WHAT'S LEFT OF MAXIMUM ARG COUNT
  Ø27D   B7              ORA     A        ; CHECK FOR Ø
  Ø27E   CA86Ø2          JZ      GNM2Ø    ; IF YES, 3 NUMBERS WERE INPUT
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 18

  Ø281              GNM15:
  Ø281   C5               PUSH    B        ; IF NOT, FILL REMAINING ARGUMENTS WITH ØFFFFH
  Ø282   2D               DCR     L
  Ø283   C281Ø2           JNZ     GNM15
  Ø286              GNM2Ø:
  Ø286   C1               POP     B        ; GET THE 3 ARGUMENTS OUT
  Ø287   D1               POP     D
  Ø288   E1               POP     H
  Ø289   CD9CØ2           CALL    HILO     ; SEE IF FIRST > = SECOND
       1            +     FALSE   GNM25    ; NO - BRANCH
  Ø28C 1 D291Ø2     +     JNC     GNM25
  Ø28F   54               MOV     D,H
  Ø29Ø   5D               MOV     E,L      ; YES - MAKE SECOND EQUAL TO THE FIRST
  Ø291              GNM25:
  Ø291   E3               XTHL             ; PUT FIRST ON STACK - GET RETURN ADDR
  Ø292   D5               PUSH    D        ; PUT SECOND ON STACK
  Ø293   C5               PUSH    B        ; PUT THIRD ON STACK
  Ø294   E5               PUSH    H        ; PUT RETURN ADDRESS ON STACK
  Ø295              GNM3Ø:
  Ø295   3D               DCR     A        ; DECREMENT RESIDUAL COUNT
  Ø296   F8               RM               ; IF NEGATIVE, PROPER RESULTS ON STACK
  Ø297   E1               POP     H        ; ELSE, GET RETURN ADDR
  Ø298   E3               XTHL             ; REPLACE TOP RESULT WITH RETURN ADDR
  Ø299   C395Ø2           JMP     GNM3Ø    ; TRY AGAIN
                    ;
                    ;
                    ;*****************************************************************
                    ;
                    ;
                    ; FUNCTION: HILO
                    ; INPUTS: DE - 16 BIT INTEGER
                    ;         HL - 16 BIT INTEGER
                    ; OUTPUTS: CARRY - Ø IF HL<DE
                    ;                - 1 IF HL>=DE
                    ; CALLS: NOTHING
                    ; DESTROYS: F/F'S
                    ; DESCRIPTION: HILO COMPARES THE 2 16 BIT INTEGERS IN HL AND DE. THE
                    ;               INTEGERS ARE TREATED AS UNSIGNED NUMBERS. THE CARRY
                    ;               BIT IS SET ACCORDING TO THE RESULT OF THE COMPARISON:
                    ;
  Ø29C              HILO:
  Ø29C   C5               PUSH    B        ; SAVE BC
  Ø29D   47               MOV     B,A      ; SAVE A IN B REGISTER
  Ø29E   E5               PUSH    H        ; SAVE HL PAIR
  Ø29F   7A               MOV     A,D      ; CHECK FOR DE = ØØØØH
  Ø2AØ   B3               ORA     E
  Ø2A1   CABDØ2           JZ      HILØ5    ; WE'RE AUTOMATICALLY DONE IF IT IS
  Ø2A4   23               INX     H        ; INCREMENT HL BY 1
  Ø2A5   7C               MOV     A,H      ; WANT TO TEST FOR Ø RESULT AFTER
  Ø2A6   B5               ORA     L        ; /INCREMENTING
  Ø2A7   CABDØ2           JZ      HILØ5    ; IF SO, HL MUST HAVE CONTAINED ØFFFFH
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 19

Ø2AA   E1              POP     H        ; IF NOT, RESTORE ORIGINAL HL
Ø2AB   D5              PUSH    D        ; SAVE DE
Ø2AC   3EFF            MVI     A,ØFFH   ; WANT TO TAKE 2'S COMPLEMENT OF DE CONTENTS
Ø2AE   AA              XRA     D
Ø2AF   57              MOV     D,A
Ø2BØ   3EFF            MVI     A,ØFFH
Ø2B2   AB              XRA     E
Ø2B3   5F              MOV     E,A
Ø2B4   13              INX     D        ; 2'S COMPLEMENT OF DE TO DE
Ø2B5   7D              MOV     A,L
Ø2B6   83              ADD     E        ; ADD HL AND DE
Ø2B7   7C              MOV     A,H
Ø2B8   8A              ADC     D        ; THIS OPERATION SETS CARRY PROPERLY
Ø2B9   D1              POP     D        ; RESTORE ORIGINAL DE CONTENTS
Ø2BA   78              MOV     A,B      ; RESTORE ORIGINAL CONTENTS OF A
Ø2BB   C1              POP     B        ; RESTORE ORIGINAL CONTENTS OF BC
Ø2BC   C9              RET              ; RETURN WITH CARRY SET AS REQUIRED
Ø2BD             HILØ5:
Ø2BD   E1              POP     H        ; IF HL CONTAINS ØFFFFH, THEN CARRY CAN
Ø2BE   78              MOV     A,B      ; /ONLY BE SET TO 1
Ø2BF   C1              POP     B        ; RESTORE ORIGINAL CONTENTS OF REGISTERS
Ø2CØ   C343Ø3          JMP     SRET     ; SET CARRY AND RETURN
                 ;
                 ;
                 ;*************************************************************************
                 ;
                 ;
                 ; FUNCTION: NMOUT
                 ; INPUTS: A - 8 BIT INTEGER
                 ; OUTPUTS: NONE
                 ; CALLS: ECHO,PRVAL
                 ; DESTROYS: A,B,C,F/F'S
                 ; DESCRIPTION: NNMOUT CONVERTS THE 8 BIT, UNSIGNED INTEGER IN THE
                 ;               A REGISTER INTO 2 ASCII CHARACTERS. THE ASCII CHARACTERS
                 ;               ARE THE ONES REPRESENTING THE 8 BITS. THESE TWO
                 ;               CHARACTERS ARE SENT TO THE CONSOLE AT THE CURRENT PRINT
                 ;               POSITION OF THE CONSOLE.
                 ;
Ø2C3             NMOUT:
Ø2C3   E5              PUSH    H        ; SAVE HL - DESTROYED BY PRVAL
Ø2C4   F5              PUSH    PSW      ; SAVE ARGUMENT
Ø2C5   ØF              RRC
Ø2C6   ØF              RRC
Ø2C7   ØF              RRC
Ø2C8   ØF              RRC              ; GET UPPER 4 BITS TO LOW 4 BIT POSITIONS
Ø2C9   E6ØF            ANI     HCHAR    : MASK OUT UPPER 4 BITS - WANT 1 HEX CHAR
Ø2CB   4F              MOV     C,A
Ø2CC   CDDEØ2          CALL    PRVAL    ; CONVERT LOWER 4 BITS TO ASCII
Ø2CF   CDF4Ø1          CALL    ECHO     ; SEND TO TERMINAL
Ø2D2   F1              POP     PSW      ; GET BACK ARGUMENT
Ø2D3   E6ØF            ANI     HCHAR    ; MASK OUT UPPER 4 BITS - WANT 1 HEX CHAR
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 2Ø

  Ø2D5   4F            MOV     C,A
  Ø2D6   CDDEØ2        CALL    PRVAL
  Ø2D9   CDF4Ø1        CALL    ECHO
  Ø2DC   E1            POP     H        ; RESTORE SAVED VALUE OF HL
  Ø2DD   C9            RET
                 ;
                 ;
                 ;*********************************************************************
                 ;
                 ;
                 ; FUNCTION;  PRVAL
                 ; INPUTS: C - INTEGER, RANGE Ø TO F
                 ; OUTPUTS: C - ASCII CHARACTER
                 ; CALLS: NOTHING
                 ; DESTROYS: B,C,H,L,F/F'S
                 ; DESCRIPTION: PRVAL CONVERTS A NUMBER IN THE RANGE Ø TO F HEX TO
                 ;               THE CORRESPONDING ASCII CHARACTER, Ø-9,A-F. PRVAL
                 ;               DOES NOT CHECK THE VALIDITY OF ITS INPUT ARGUMENT.
                 ;
  Ø2DE           PRVAL:
  Ø2DE   21BFØ3        LXI     H,DIGTB; ADDRESS OF TABLE
  Ø2E1   Ø6ØØ          MVI     B,Ø     ; CLEAR HIGH ORDER BITS OF BC
  Ø2E3   Ø9            DAD     B       ; ADD DIGIT VALUE TO HL ADDRESS
  Ø2E4   4E            MOV     C,M     ; FETCH CHARACTER FROM MEMORY
  Ø2E5   C9            RET
                 ;

                 ;*********************************************************************
                 ;
                 ;
                 ; FUNCTION: REGDS
                 ; INPUTS: NONE
                 ; OUTPUTS: NONE
                 ; CALLS: ECHO,NMOUT,ERROR,CROUT
                 ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                 ; DESCRIPTION: REGDS DISPLAYS THE CONTENTS OF THE REGISTER SAVE
                 ;               LOCATIONS, IN FORMATTED FORM, ON THE CONSOLE. THE
                 ;               DISPLAY IS DRIVEN FROM A TABLE, RTAB, WHICH CONTAINS
                 ;               THE REGISTER'S PRINT SYMBOL, SAVE LOCATION ADDRESS,
                 ;               AND LENGTH (8 OR 16 BITS).
                 ;
  Ø2E6           REGDS:
  Ø2E6   21CFØ3        LXI     H,RTAB  ; LOAD HL WITH ADDRESS OF START OF TABLE
  Ø2E9           REGØ5:
  Ø2E9   4E            MOV     C,M     ; GET PRINT SYMBOL OF REGISTER
  Ø2EA   79            MOV     A,C
  Ø2EB   B7            ORA     A       ; TEST FOR Ø - END OF TABLE
  Ø2EC   C2F3Ø2        JNZ     REG1Ø   ; IF NOT END, BRANCH
  Ø2EF   CDEEØ1        CALL    CROUT   ; ELSE, CARRIAGE RETURN/LINE FEED TO END
  Ø2F2   C9            RET             ; /DISPLAY
  Ø2F3           REG1Ø:
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 21

  Ø2F3   CDF4Ø1        CALL    ECHO     ; ECHO CHARACTER
  Ø2F6   ØE3D          MVI     C,'='
  Ø2F8   CDF4Ø1        CALL    ECHO     ; OUTPUT EQUALS SIGN, I.E. A=
  Ø2FB   23            INX     H        ; POINT TO START OF SAVE LOCATION ADDRESS
  Ø2FC   5E            MOV     E,M      ; GET LSP OF SAVE LOCATION ADDRESS TO E
  Ø2FD   1613          MVI     D,DATA SHR 8    ; PUT MSP OF SAVE LOC ADDRESS INTO D
  Ø2FF   23            INX     H        ; POINT TO LENGTH FLAG
  Ø3ØØ   1A            LDAX    D        ; GET CONTENTS OF SAVE ADDRESS
  Ø3Ø1   CDC3Ø2        CALL    NMOUT    ; DISPLAY ON CONSOLE
  Ø3Ø4   7E            MOV     A,M      ; GET LENGTH FLAG
  Ø3Ø5   B7            ORA     A        ; SET SIGN F/F
  Ø3Ø6   CAØEØ3        JZ      REG15    ; IF Ø, REGISTER IS 8 BITS
  Ø3Ø9   1B            DCX     D        ; ELSE, 16 BIT REGISTER SO MORE TO DISPLAY
  Ø3ØA   1A            LDAX    D        ; GET LOWER 8 BITS
  Ø3ØB   CDC3Ø2        CALL    NMOUT    ; DISPLAY THEM
  Ø3ØE                 REG15:
  Ø3ØE   ØE2Ø          MVI     C,''
  Ø31Ø   CDF4Ø1        CALL    ECHO
  Ø313   23            INX     H        ; POINT TO START OF NEXT TABLE ENTRY
  Ø314   C3E9Ø2        JMP     REGØ5    ; DO NEXT REGISTER
                       ;
                       ;
                       ;*****************************************************************
                       ;
                       ;
                       ; FUNCTION: RGADR
                       ; INPUTS: C - CHARACTER DENOTING REGISTER
                       ; OUTPUTS: BC - ADDRESS OF ENTRY IN RTAB CORRESPONDING TO REGISTER
                       ; CALLS: ERROR
                       ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                       ; DESCRIPTION: RGADR TAKES A SINGLE CHARACTER AS INPUT. THIS CHARACTER
                       ;              DENOTES A REGISTER. RGADR SEARCHES THE TABLE RTAB
                       ;              FOR A MATCH ON THE INPUT ARGUMENT. IF ONE OCCURS,
                       ;              RGADR RETURNS THE ADDRESS OF THE ADDRESS OF THE
                       ;              SAVE LOCATION CORRESPONDING TO THE REGISTER. THIS
                       ;              ADDRESS POINTS INTO RTAB. IF NO MATCH OCCURS; THEN
                       ;              THE REGISTER IDENTIFIER IS ILLEGAL AND CONTROL IS
                       ;              PASSED TO THE ERROR ROUTINE.
                       ;
  Ø317                 RGADR:
  Ø317   21CFØ3        LXI     H,RTAB   ; HL GETS ADDRESS OF TABLE START
  Ø31A   11Ø3ØØ        LXI     D,RTABS  ; DE GET SIZE OF A TABLE ENTRY
  Ø31D                 RGAØ5:
  Ø31D   7E            MOV     A,M      ; GET REGISTER IDENTIFIER
  Ø31E   B7            ORA     A        ; CHECK FOR TABLE END (IDENTIFIER IS Ø)
  Ø31F   CAØDØ2        JZ      ERROR    ; IF AT END OF TABLE, ARGUMENT IS ILLEGAL
  Ø322   B9            CMP     C        ; ELSE, COMPARE TABLE ENTRY AND ARGUMENT
  Ø323   CA2AØ3        JZ      RGA1Ø    ; IF EQUAL, WE'VE FOUND WHAT WE'RE LOOKING FOR
  Ø326   19            DAD     D        ; ELSE, INCREMENT TABLE POINTER TO NEXT ENTRY
  Ø327   C31DØ3        JMP     RGAØ5    ; TRY AGAIN
  Ø32A                 RGA1Ø:
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 22

 Ø32A   23          INX     H        ; IF A MATCH, INCREMENT TABLE POINTER TO
 Ø32B   44          MOV     B,H      ; /SAVE LOCATION ADDRESS
 Ø32C   4D          MOV     C,L      ; RETURN THIS VALUE
 Ø32D   C9          RET
                ;
                ;
                ;*****************************************************************
                ;
                ;
                ; FUNCTION: RSTTF
                ; INPUTS: NONE
                ; OUTPUTS: NONE
                ; CALLS: NOTHING
                ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                ; DESCRIPTION: RSTTF RESTORES ALL CPU REGISTER, FLIP/FLOPS, STACK
                ;              POINTER AND PROGRAM COUNTER FROM THEIR RESPECTIVE
                ;              SAVE LOCATIONS IN MEMORY. THE ROUTINE THEN TRANSFERS
                ;              CONTROL TO THE LOCATION SPECIFIED BY THE PROGRAM
                ;              COUNTER (I.E. THE RESTORED VALUE). THE ROUTINE
                ;              EXITS WITH THE INTERRUPTS ENABLED.
                ;
 Ø32E           RSTTF:
 Ø32E   F3 10        DI              ; DISABLE INTERRUPTS WHILE RESTORING THINGS
§Ø32F   21ED13       LXI     H,MSTAK ; SET MONITOR STACK POINTER TO START OF STACK
 Ø332   F9           SPHL
 Ø333   D1           POP     D       ; START ALSO END OF REGISTER SAVE AREA
 Ø334   C1           POP     B
 Ø335   F1 10        POP     PSW
§Ø336   2AF713       LHLD    SSAVE   ; RESTORE USER STACK POINTER
 Ø339   F9 10        SPHL
§Ø33A   2AF513       LHLD    PSAVE
 Ø33D   E5           PUSH    H       ; PUT USER RETURN ADDRESS ON USER STACK
§Ø33E   2AF313       LHLD    LSAVE   ; RESTORE HL REGISTERS
 Ø341   FB 10        EI              ; ENABLE INTERRUPTS NOW
 Ø342   C9           RET             ; JUMP TO RESTORED PC LOCATION
                ;
                ;
                ;*****************************************************************
                ;
                ;
                ; FUNCTION: SRET
                ; INPUTS: NONE
                ; OUTPUTS: CARRY = 1
                ; CALLS: NOTHING
                ; DESTROYS: CARRY
                ; DESCRIPTION: SRET IS JUMPED TO BY ROUTINES WISHING TO RETURN SUCCESS.
                ;              SRET SETS THE CARRY TRUE AND THEN RETURNS TO THE
                ;              CALLER OF THE ROUTINE INVOKING SRET.
                ;
 Ø343           SRET:
 Ø343   37           STC             ; SET CARRY TRUE
```

```
8080 MACRO ASSEMBLER, VER 2.3  ERRORS = 0 PAGE 23

 0344   C9              RET              ; RETURN APPROPRIATELY
                  ;
                  ;
                  ;*****************************************************************
                  ;
                  ;
                  ; FUNCTION: STHF0
                  ; INPUTS: DE - 16 BIT ADDRESS OF BYTE TO BE STORED INTO
                  ; OUTPUTS: NONE
                  ; CALLS: STHLF
                  ; DESTROYS: A,B,C,H,L,F/F'S
                  ; DESCRIPTION: STHF0 CHECKS THE HALF BYTE FLAG IN TEMP TO SEE IF
                  ;               IT IS SET TO LOWER. IF SO, STHF0 STORES A 0 TO
                  ;               PAD OUT THE LOWER HALF OF THE ADDRESSED BYTE;
                  ;               OTHERWISE, THE ROUTINE TAKES NO ACTION.
                  ;
 0345        10   STHF0:
 0345   3AF913          LDA     TEMP     ; GET HALF BYTE FLAG
 0348   B7              ORA     A        ; SET F/F'S
 0349   C0              RNZ              ; IF SET TO UPPER, DON'T DO ANYTHING
 034A   0E00            MVI     C,0      ; ELSE, WANT TO STORE THE VALUE 0
 034C   CD5003          CALL    STHLF    ; DO IT
 034F   C9              RET
                  ;
                  ;
                  ;*****************************************************************
                  ;
                  ;
                  ; FUNCTION: STHLF
                  ; INPUTS: C - 4 BIT VALUE TO BE STORED IN HALF BYTE
                  ;         DE - 16 BIT ADDRESS OF BYTE TO BE STORED INTO
                  ; OUTPUTS: NONE
                  ; CALLS: NOTHING
                  ; DESTROYS: A,B,C,H,L,F/F'S
                  ; DESCRIPTION: STHLF TAKES THE 4 BIT VALUE IN C AND STORES IT IN
                  ;               HALF OF THE BYTE ADDRESSED BY REGISTERS DE. THE
                  ;               HALF BYTE USED (EITHER UPPER OR LOWER) IS DENOTED
                  ;               BY THE VALUE OF THE FLAG IN TEMP. STHLF ASSUMES
                  ;               THAT THIS FLAG HAS BEEN PREVIOUSLY SET
                  ;               (NOMINALLY BY ICMD).
                  ;
 0350             STHLF:
 0350   D5              PUSH    D
 0351   E1              POP     H        ; MOVE ADDRESS OF BYTE INTO HL
 0352   79              MOV     A,C      ; GET VALUE
 0353   E60F            ANI     0FH      ; FORCE TO 4 BIT LENGTH
 0355   4F        10    MOV     C,A      ; PUT VALUE BACK
 0356   3AF913          LDA     TEMP     ; GET HALF BYTE FLAG
 0359   B7              ORA     A        ; CHECK FOR LOWER HALF
 035A   C2630 3         JNZ     STH05    ; BRANCH IF NOT
 035D   7E              MOV     A,M      ; ELSE, GET BYTE
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 24

  Ø35E   E6FØ          ANI     ØFØH     ; CLEAR LOWER 4 BITS
  Ø36Ø   B1            ORA     C        ; OR IN VALUE
  Ø361   77            MOV     M,A      ; PUT BYTE BACK
  Ø362   C9            RET
  Ø363             STHØ5:
  Ø363   7E            MOV     A,M      ; IF UPPER HALF, GET BYTE
  Ø364   E6ØF          ANI     ØFH      ; CLEAR UPPER 4 BITS
  Ø366   47            MOV     B,A      ; SAVE BYTE IN B
  Ø367   79            MOV     A,C      ; GET VALUE
  Ø368   ØF            RRC
  Ø369   ØF            RRC
  Ø36A   ØF            RRC
  Ø36B   ØF            RRC              ; ALIGN TO UPPER 4 BITS
  Ø36C   BØ            ORA     B        ; OR IN ORIGINAL LOWER 4 BITS
  Ø36D   77            MOV     M,A      ; PUT NEW CONFIGURATION BACK
  Ø36E   C9            RET
                   ;
                   ;
                   ;*****************************************************************
                   ;
                   ;
                   ; FUNCTION: VALDG
                   ; INPUTS: C - ASCII CHARACTER
                   ; OUTPUTS: CARRY - 1 IF CHARACTER REPRESENTS VALID HEX DIGIT
                   ;                 - Ø OTHERWISE
                   ; CALLS: NOTHING
                   ; DESTROYS: A,F/F'S
                   ; DESCRIPTION: VALDG RETURNS SUCCESS IF ITS INPUT ARGUMENT IS
                   ;               AN ASCII CHARACTER REPRESENTING A VALID HEX DIGIT
                   ;                    `-F), AND FAILURE OTHERWISE:
                   ;
  Ø36F             VALDG:
  Ø36F   79            MOV     A,C
  Ø37Ø   FE3Ø          CPI     ´Ø´      ; TEST CHARACTER AGAINST ´Ø´
  Ø372   FA18Ø2        JM      FRET     ; IF ASCII CODE LESS, CANNOT BE VALID DIGIT
  Ø375   FE39          CPI     ´9´      ; ELSE, SEE IF IN RANGE ´Ø´-´9´
  Ø377   FA43Ø3        JM      SRET     ; CODE BETWEEN ´Ø´and ´9´
  Ø37A   CA43Ø3        JZ      SRET     ; CODE EQUAL ´9´
  Ø37D   FE41          CPI     ´A´      ; NOT A DIGIT - TRY FOR A LETTER
  Ø37F   FA18Ø2        JM      FRET     ; NO - CODE BETWEEN ´9´ AND ´A´
  Ø382   FE47          CPI     ´G´
  Ø384   F218Ø2        JP      FRET     ; NO - CODE GREATER THAN ´F´
  Ø387   C343Ø3        JMP     SRET     ; OKAY - CODE IS ´A´ TO ´F´, INCLUSIVE
                   ;
                   ;
                   ;*****************************************************************
                   ;
                   ;
                   ; FUNCTION: VALDL
                   ; INPUTS: C - CHARACTER
                   ; OUTPUTS: CARRY - 1 IF INPUT ARGUMENT VALID DELIMITER
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 25

                    ;                    - Ø OTHERWISE
                    ; CALLS: NOTHING
                    ; DESTROYS: A,F/F'S
                    ; DESCRIPTION: VALDL RETURNS SUCCESS IF ITS INPUT ARGUMENT IS A VALID
                    ;               DELIMITER CHARACTER (SPACE, COMMA, CARRIAGE RETURN) AND
                    ;               FAILURE OTHERWISE.
                    ;
 Ø38A               VALDL:
 Ø38A   79                MOV     A,C
 Ø38B   FE2C              CPI     ´,´       ; CHECK FOR COMMA
 Ø38D   CA43Ø3            JZ      SRET
 Ø39Ø   FEØD              CPI     CR        ; CHECK FOR CARRIAGE RETURN
 Ø392   CA43Ø3            JZ      SRET
 Ø395   FE2Ø              CPI     ´ ´       ; CHECK FOR SPACE
 Ø397   CA43Ø3            JZ      SRET
 Ø39A   C318Ø2            JMP     FRET      ; ERROR IF NONE OF THE ABOVE
                    ;
                    ;
                    ;*****************************************************************
                    ;
                    ;
                    ;                          MONITOR TABLES
                    ;
                    ;
                    ;*****************************************************************
                    ;
                    ;
 Ø39D               SGNON:                          ; SIGNON MESSAGE
 Ø39D   ØDØA4D43          DB      CR,LF,´MCS-8Ø KIT´,CR,LF
 Ø3A1   532D383Ø
 Ø3A5   2Ø4B4954
 Ø3A9   ØDØA
 ØØØE               LSGNON          EQU     S-SGNON ; LENGTH OF SIGNON MESSAGE
                    ;
 Ø3AB               CADR:                           ; TABLE OF ADDRESSES OF COMMAND ROUTINES
 Ø3AB   ØØØØ              DW      Ø         ; DUMMY
 Ø3AD   41Ø1              DW      XCMD
 Ø3AF   1DØ1              DW      SCMD
 Ø3B1   FDØØ              DW      MCMD
 Ø3B3   B3ØØ              DW      ICMD
 Ø3B5   95ØØ              DW      GCMD
 Ø3B7   5EØØ              DW      DCMD
                    ;
 Ø3B9               CTAB:                           ; TABLE OF VALID COMMAND CHARACTERS
 Ø3B9   44                DB      ´D´
 Ø3BA   47                DB      ´G´
 Ø3BB   49                DB      ´I´
 Ø3BC   4D                DB      ´M´
 Ø3BD   53                DB      ´S´
 Ø3BE   58                DB      ´X´
 ØØØ6               NCMDS EQU       S-CTAB  ; NUMBER OF VALID COMMANDS
```

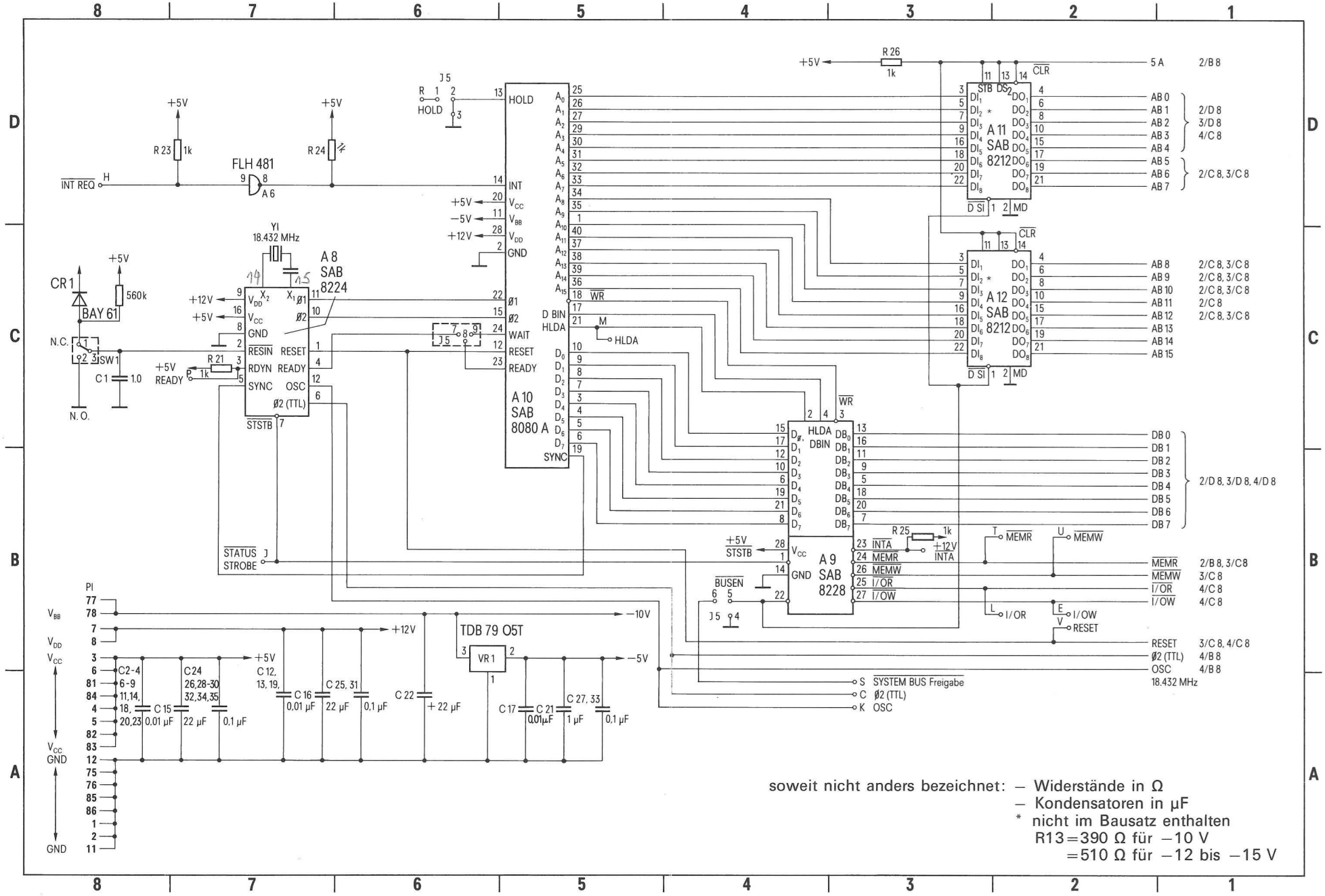
```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 26

                    ;
Ø3BF                DIGTB:                          ; TABLE OF PRINT VALUES OF HEX DIGITS
Ø3BF   3Ø                 DB        'Ø'
Ø3CØ   31                 DB        '1'
Ø3C1   32                 DB        '2'
Ø3C2   33                 DB        '3'
Ø3C3   34                 DB        '4'
Ø3C4   35                 DB        '5'
Ø3C5   36                 DB        '6'
Ø3C6   37                 DB        '7'
Ø3C7   38                 DB        '8'
Ø3C8   39                 DB        '9'
Ø3C9   41                 DB        'A'
Ø3CA   42                 DB        'B'
Ø3CB   43                 DB        'C'
Ø3CC   44                 DB        'D'
Ø3CD   45                 DB        'E'
Ø3CE   46                 DB        'F'
                    ;
Ø3CF                RTAB:                     ; TABLE OF REGISTER INFORMATION
Ø3CF   41                 DB        'A'       ; REGISTER IDENTIFIER
Ø3DØ   F2                 DB        ASAVE AND ØFFH  ; ADDRESS OF REGISTER SAVE LOCATION
Ø3D1   ØØ                 DB        Ø         ; LENGTH FLAG - Ø=8 BITS, 1=16 BITS
ØØØ3                RTABS EQU       S-RTAB    ; SIZE OF AN ENTRY IN THIS TABLE
Ø3D2   42                 DB        'B'
Ø3D3   FØ                 DB        BSAVE AND ØFFH
Ø3D4   ØØ                 DB        Ø
Ø3D5   43                 DB        'C'
Ø3D6   EF                 DB        CSAVE AND ØFFH
Ø3D7   ØØ                 DB        Ø
Ø3D8   44                 DB        'D'
Ø3D9   EE                 DB        DSAVE AND ØFFH
Ø3DA   ØØ                 DB        Ø
Ø3DB   45                 DB        'E'
Ø3DC   ED                 DB        ESAVE AND ØFFH
Ø3DD   ØØ                 DB        Ø
Ø3DE   46                 DB        'F'
Ø3DF   F1                 DB        FSAVE AND ØFFH
Ø3EØ   ØØ                 DB        Ø
Ø3E1   48                 DB        'H'
Ø3E2   F4                 DB        HSAVE AND ØFFH
Ø3E3   ØØ                 DB        Ø
Ø3E4   4C                 DB        'L'
Ø3E5   F3                 DB        LSAVE AND ØFFH
Ø3E6   ØØ                 DB        Ø
Ø3E7   4D                 DB        'M'
Ø3E8   F4                 DB        HSAVE AND ØFFH
Ø3E9   Ø1                 DB        1
Ø3EA   5Ø                 DB        'P'
Ø3EB   F6                 DB        PSAVE+1 AND ØFFH
Ø3EC   Ø1                 DB        1
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 27

  Ø3ED   53            DB      'S'
  Ø3EE   F8            DB      SSAVE+1 AND ØFFH
  Ø3EF   Ø1            DB      1
  Ø3FØ   ØØ            DB      Ø        ; END OF TABLE MARKERS
  Ø3F1   ØØ            DB      Ø
                 ;
  Ø3FA                 ORG     BRTAB
                 ;
  Ø3FA   C3E3Ø1        JMP     CO       ; BRANCH TABLE FOR USER ACCESSIBLE ROUTINES
  Ø3FD   C3DØØ1        JMP     CI
                 ;
                 ;
                 ;*****************************************************************
                 ;
                 ;
  13ØØ                 ORG     DATA
  13ED                 ORG     REGS     ; ORG TO REGISTER SAVE - STACK GOES IN HERE
                 ;
  13ED           MSTAK         EQU      S        ; START OF MONITOR STACK
  13ED   ØØ      ESAVE:        DB       Ø        ; E REGISTER SAVE LOCATION
  13EE   ØØ      DSAVE:        DB       Ø        ; D REGISTER SAVE LOCATION
  13EF   ØØ      CSAVE:        DB       Ø        ; C REGISTER SAVE LOCATION
  13FØ   ØØ      BSAVE:        DB       Ø        ; B REGISTER SAVE LOCATION
  13F1   ØØ      FSAVE:        DB       Ø        ; FLAGS SAVE LOCATION
  13F2   ØØ      ASAVE:        DB       Ø        ; A REGISTER SAVE LOCATION
  13F3   ØØ      LSAVE:        DB       Ø        ; L REGISTER SAVE LOCATION
  13F4   ØØ      HSAVE:        DB       Ø        ; H REGISTER SAVE LOCATION
  13F5   ØØØØ    PSAVE:        DW       Ø        ; PGM COUNTER SAVE LOCATION
  13F7   ØØØØ    SSAVE:        DW       Ø        ; USER STACK POINTER SAVE LOCATION
  13F9   ØØ      TEMP:         DB       Ø        ; TEMPORARY MONITOR CELL
                 ;
  13FD                 ORG     BRLOC             ; ORG TO USER BRANCH LOCATION
                 ;
  ØØØ3           USRBR:        DS       3        ; BRANCH GOES IN HERE
                 ;
                 ;
                       END
NO PROGRAM ERRORS
```

```
8Ø8Ø MACRO ASSEMBLER, VER 2.3  ERRORS = Ø PAGE 28

                    SYMBOL TABLE

* Ø1

A       ØØØ7      ASAVE   13F2      B       ØØØØ      BRCHR   ØØ1B
BREAK   Ø1BD      BRLOC   13FD      BRTAB   Ø3FA      BSAVE   13FØ
C       ØØØ1      CADR    Ø3AB      CI      Ø1DØ      CMD     ØØ27
CNCTL   ØØFB      CNIN    ØØFA      CNOUT   ØØFA      CNVBN   Ø1DA
CO      Ø1E3      CONST   ØØFB      CR      ØØØD      CROUT   Ø1EE
CSAVE   13EF      CTAB    Ø3B9      D       ØØØ2      DATA    13ØØ
DCMØ5   ØØ65      DCM1Ø   ØØ7Ø      DCM12   ØØ85      DCM15   ØØ8B
DCMD    ØØ5E      DIGTB   Ø3BF      DSAVE   13EE      E       ØØØ3
ECHØ5   Ø1FD      ECH1Ø   Ø2ØB      ECHO    Ø1F4      ERROR   Ø2ØD
ESAVE   13ED      ESC     ØØ1B      FALSE   ØF9C      FRET    Ø218
FSAVE   13F1      GCMØ5   ØØAA      GCM1Ø   ØØBØ      GCMD    ØØ95
GETCH   Ø21B      GETCM   ØØ2B      GETHX   Ø222      GETNM   Ø257
GHXØ5   Ø228      GHX1Ø   Ø241      GNMØ5   Ø25E      GNM1Ø   Ø273
GNM15   Ø281      GNM2Ø   Ø286      GNM25   Ø291      GNM3Ø   Ø295
GO      ØØØ8 *    GTCØ3   ØØ3B      GTCØ5   ØØ48      GTC1Ø   ØØ54
H       ØØØ4      HCHAR   ØØØF      HILØ5   Ø2BD      HILO    Ø29C
HSAVE   13F4      ICMØ5   ØØBE      ICM1Ø   ØØE6      ICM2Ø   ØØEE
ICM25   ØØF4      ICMD    ØØB3      INVRT   ØØFF      L       ØØØ5
LF      ØØØA      LOWER   ØØØØ *    LSAVE   13F3      LSGNO   ØØØE
M       ØØØ6      MCMØ5   Ø1Ø5      MCMD    ØØFD      MODE    ØØCF
MSGL    ØØ22      MSTAK   13ED      NCMDS   ØØØ6      NEWLN   ØØØF
NMOUT   Ø2C3      PRTYØ   ØØ7F      PRVAL   Ø2DE      PSAVE   13F5
PSW     ØØØ6      RBR     ØØØ2      REGØ5   Ø2E9      REG1Ø   Ø2F3
REG15   Ø3ØE      REGDS   Ø2E6      REGS    13ED      RGAØ5   Ø31D
RGA1Ø   Ø32A      RGADR   Ø317      RSTTF   Ø32E      RSTU    ØØ38
RTAB    Ø3CF      RTABS   ØØØ3      SCMØ5   Ø122      SCM1Ø   Ø12D
SCM15   Ø13D      SCMD    Ø11D      SGNON   Ø39D      SP      ØØØ6
SRET    Ø343      SSAVE   13F7      STHØ5   Ø363      STHFØ   Ø345
STHLF   Ø35Ø      TEMP    13F9      TERM    ØØ1B      TRDY    ØØØ1
TRUE    ØF9F      UPPER   ØØFF      USRBR   13FD      VALDG   Ø36F
VALDL   Ø38A      XCMØ5   Ø154      XCM1Ø   Ø163      XCM15   Ø17Ø
XCM18   Ø17B      XCM2Ø   Ø194      XCM25   Ø1AB      XCM27   Ø1AC
XCM3Ø   Ø1B4      XCMD    Ø141

* Ø2
* Ø3
* Ø4
* Ø5
* Ø6
* Ø7
* Ø8
* Ø9
* 1Ø
* 11
* 12
* 13
..L
```

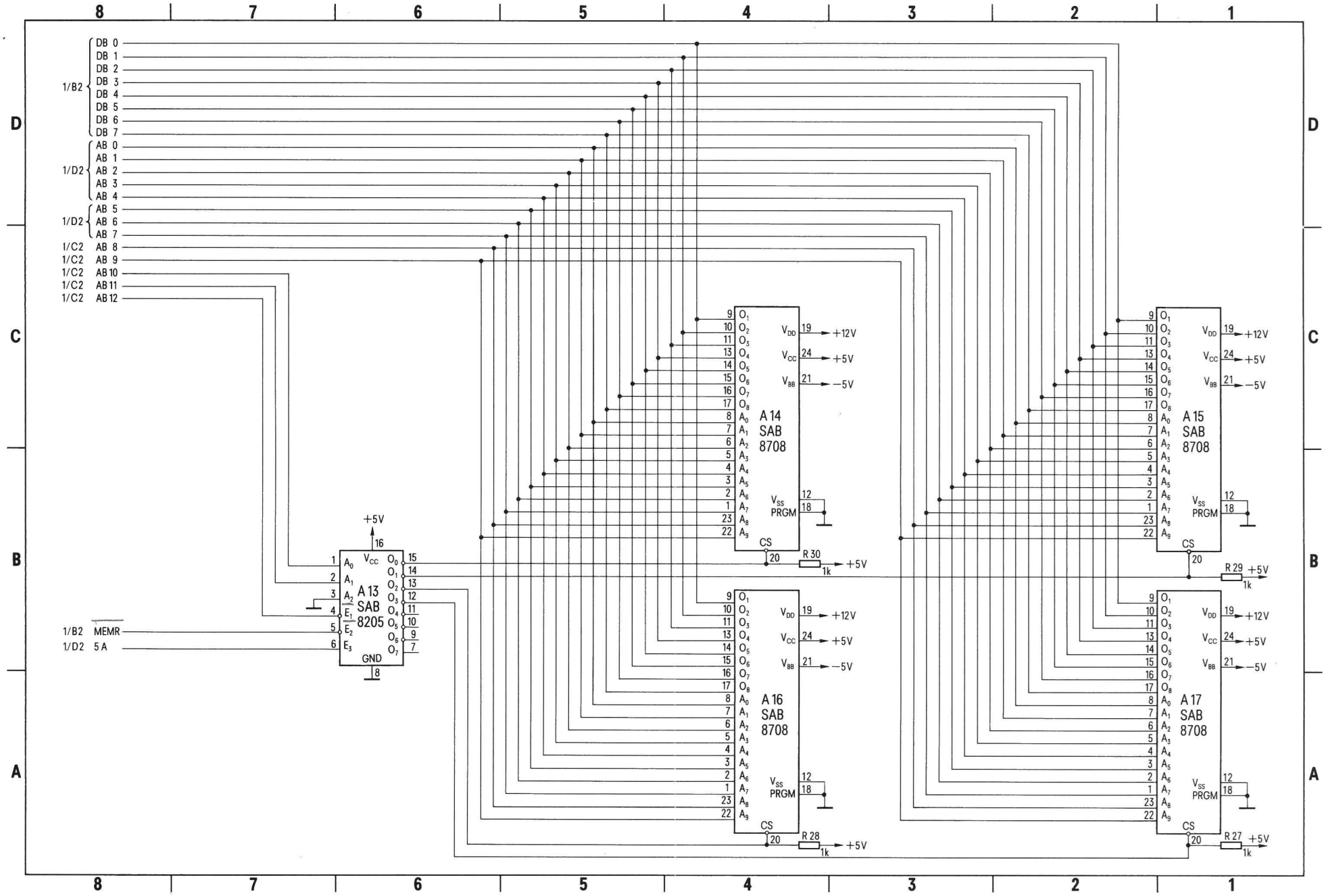
7. Gesamtschaltung des SIKIT-DK/8080-Computers

Bild 23 Mikroprozessor



VX

Bild 24
PROM-Bereich



VX

Zu Bild 26

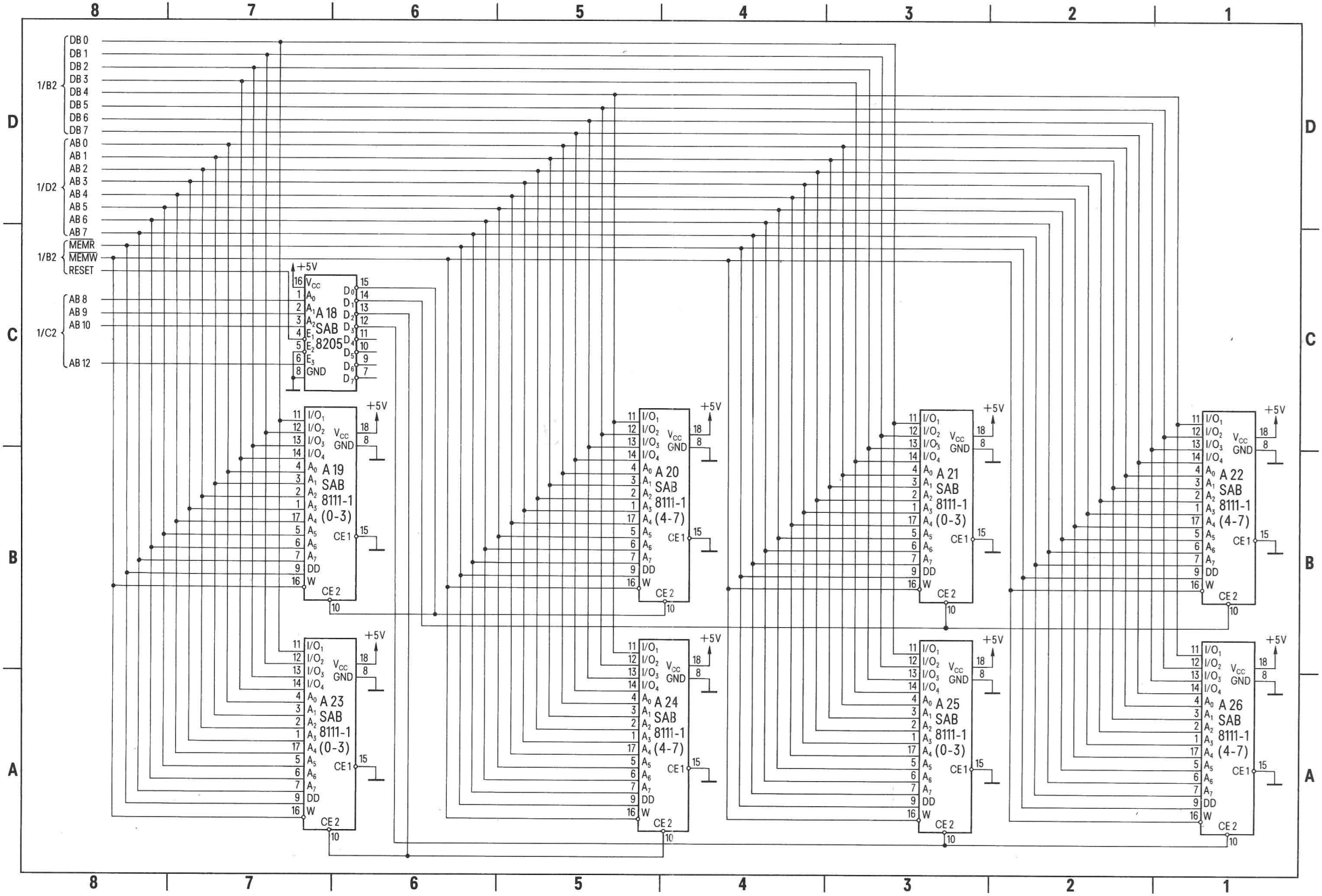
Je nach Anwenderkonfiguration sind folgende Drahtbrücken einzufügen:
(Zu finden im Schaltbild unter Suchraster 1D, 1C, 1B, 1A)

CRT	TTY	TTL	Datenübertragungsrate	
23 mit 24	23 mit 26	23 mit 25	–	–
18 mit 17	18 mit 19	18 mit 17	29 mit 31	4800
10 mit 9	11 mit 10	13 mit 12	29 mit 32	2400
14 mit 13	14 mit 13	2 mit 3	29 mit 33	1200
2 mit 3	1 mit 2	21 mit 20	29 mit 34	600
6 mit 8	7 mit 8		29 mit 35	300
27 mit 28	15 mit 16		29 mit 36	150
22 mit 21	22 mit 21		29 mit 37	75/110 (Falls Punkt 4 mit 5 verbunden wurde)

invertiert 1/0 nicht invertiert 1/0

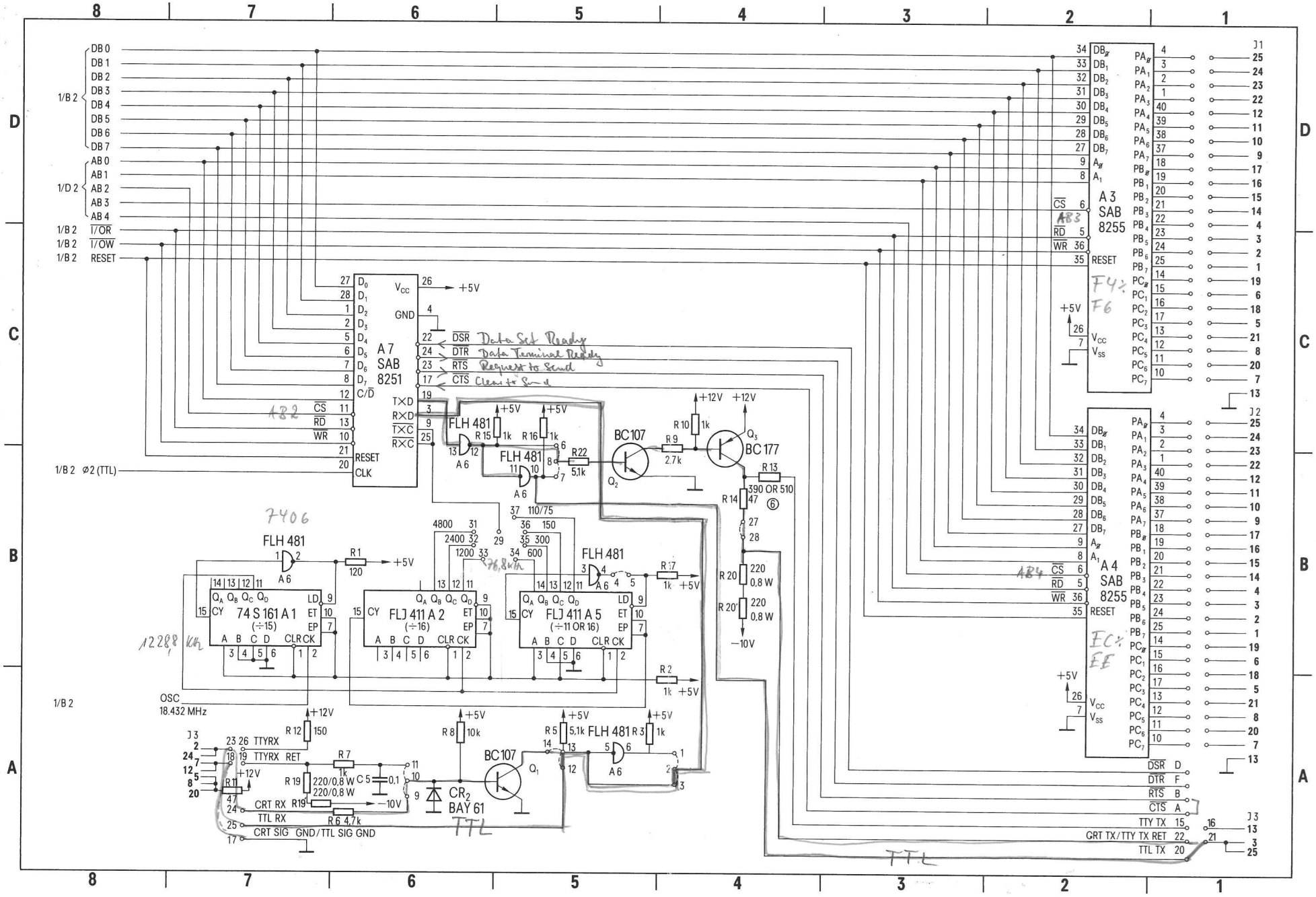
VX

Bild 25
RAM-Bereich



VX

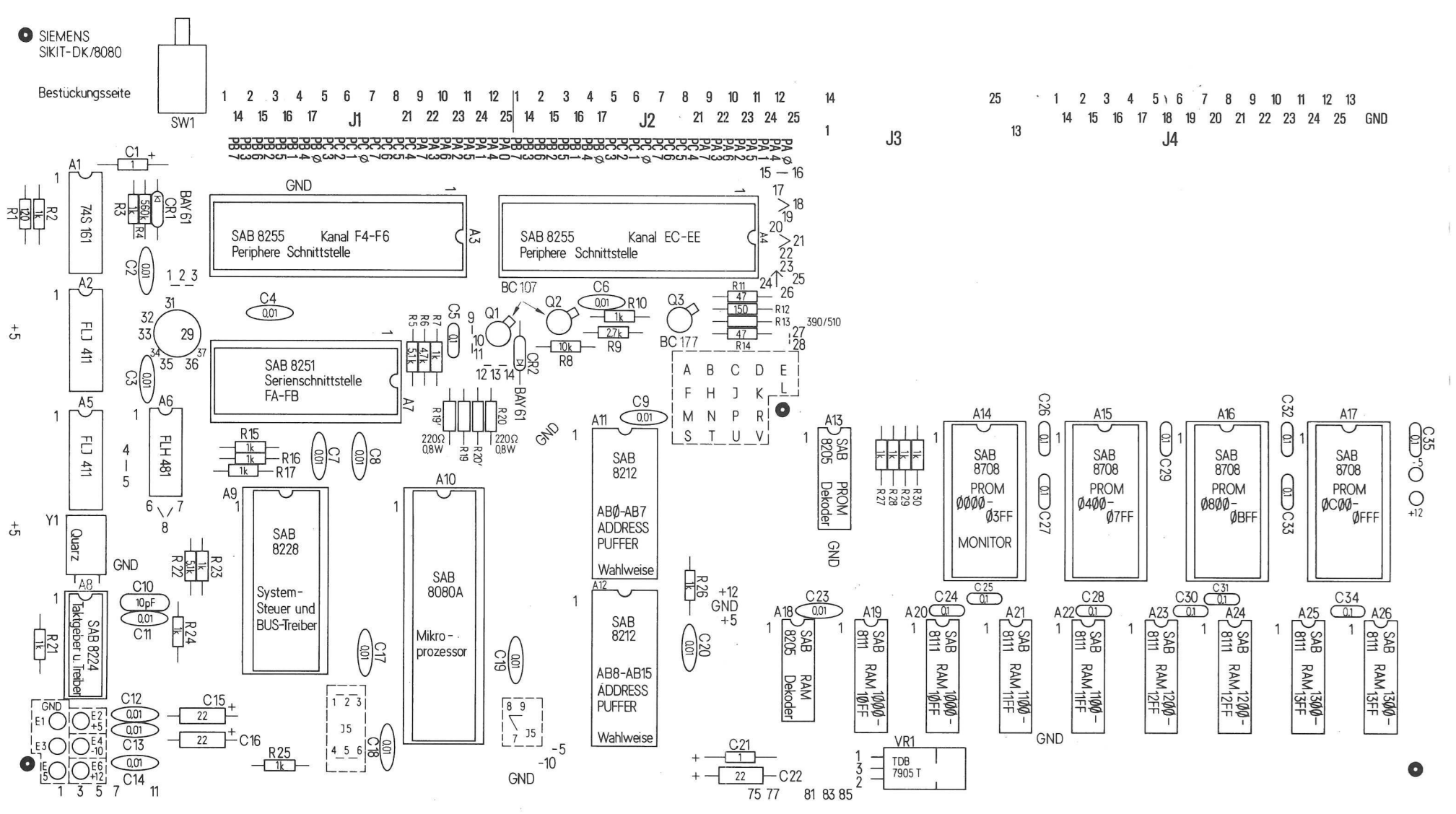
Bild 26
E/A-Baustein-Konfiguration



VX

97
Vx

Bild 27
Gesamtbestückungsplan
(Widerstände in Ω, Kondensatoren in µF, soweit nicht anders bezeichnet)



Zweigniederlassungen in der Bundesrepublik Deutschland und Berlin (West)

1000 Berlin 61
Schöneberger Straße 2-4
Postanschrift:
1000 Berlin 11, Postfach 11 05 60
Tel. 2 55-1, Telex 1 83 766

2800 Bremen 1
Contrescarpe 72
Postfach 127
Tel. 364-1
Telex 245451

4600 Dortmund 1
Märkische Straße 8-14
Postfach 658
Tel. 54 90-1
Telex 8 22 312

4000 Düsseldorf 1
Lahnweg 10
Postfach 1115
Tel. 30 30-1
Telex 8 581301

4300 Essen 1
Kruppstraße 16
Postfach 22
Tel. 2013-1
Telex 857437

6000 Frankfurt (Main) 1
Gutleutstraße 31
Postfach 4183
Tel. 2 62-1
Telex 414131

2000 Hamburg 1
Lindenplatz 2
Postfach 10 56 09
Tel. 2 82–1
Telex 21 62 721

3000 Hannover 1
Am Maschpark 1
Postfach 53 29
Tel. 199-1
Telex 9 22 333

5000 Köln 30
Franz-Geuer-Str. 10
Postfach 30 11 66
Tel. 5 76-1
Telex 8 881005

6800 Mannheim 1
N 7.18
Postfach 20 24
Tel. 2 96-1
Telex 4 62 261

8000 München 80
Richard-Strauss-Straße 76
Postanschrift:
8000 München 2
Postfach 20 2109
Tel. 92 21-1
Telex 5 29 421

8500 Nürnberg 1
Von-der-Tann-Straße 30
Postfach 24 29
Tel. 6 54-1
Telex 6 22 251

6600 Saarbrücken 3
Martin-Luther-Straße 25
Postfach 359
Tel. 30 08-1
Telex 4 421431

7000 Stuttgart 1
Geschwister-Scholl-Straße 24
Postfach 120
Tel. 20 76-1
Telex 7 23 941

Siemens-Landesgesellschaften und -Vertretungen

Europa

Belgien
Siemens Société Anonyme
Chaussée de Charleroi 116
B-1060 Bruxelles
Tel. 5 37 3100, Telex 21347

Bulgarien
RUEN
Technisches Beratungsbüro
der Siemens AG
uliza Nikolai Gogol 5/
Boulevard Lenin
BG-1504 Sofia 4
Tel. 45 70 82, Telex 22 763

Dänemark
Siemens Aktieselskab
Borupvang 3
DK-2750 Ballerup
Tel. 65 65 65, Telex 35 313

Finnland
Siemens Osakeyhtiö
Mikonkatu 8
SF-00101 Helsinki 10
(PL 8)
Tel. 107 14, Telex 12 465

Frankreich
Siemens S.A.
B.P. 109
F-93203 Saint-Denis CEDEX 1
Tel. 8 20 61 20, Tx. 62 0853

Griechenland
Siemens Hellas E.A.E.
Voulis 7
Athen 125 (P.O.B. 601)
Tel. 32 93-1, Telex 216 291

Großbritannien
Siemens Ltd.
Great West House,
Great West Road
Brentford TW8 9DG
Tel. 5 68 91 33, Telex 23 176

Irland
Siemens Ltd.
8, Raglan Road
Dublin 4
Tel. 68 47 27, Telex 5341

Island
Smith & Norland H/F
Nóatún 4,
Reykjavik (P.O.B. 519)
Tel. 2 83 22, Telex 20 55

Italien
Siemens Elettra S.p.A.
Via Vittor Pisani, 20
I-20124 Milano
(Casella Postale 4183)
Tel. 62 48, Telex 36 261

Jugoslawien
Generalexport
Masarikava 5/XV
YU-11000 Beograd
(YU-1101 Beograd
Poštanski fah 223)
Tel. 6 84-866, Telex 11 287

Luxemburg
Siemens Société Anonyme
Rue Glesener 17
Luxembourg (P.B. 1701)
Tel. 49 711-1, Telex 3430

Niederlande
Siemens Nederland N.V.
Prinses Beatrixlaan 26
Den Haag 2077
(Postbus 1068)
Tel. 78 27 82, Telex 31 373

Norwegen
Siemens A/S
Østre Aker Vei 90
N-Oslo 5
(Postboks 10, Veitvet)
Tel. 15 30 90, Telex 18 477

Österreich
Siemens Aktiengesellschaft
Österreich
A-1030 Wien,
Apostelgasse 12
(A-1031 Wien, Postfach 326)
Tel. 72 93-0, Telex 11 866

Polen
PHZ Transactor S.A.
PL-00-950 Warszawa
(P.O.B. 30)
Tel. 49 72 62, Telex 813 288

Portugal
Siemens S.A.R.L.
Av. Almirante Reis, 65
Lisboa-1 (Apartado 1380)
Tel. 53 88 05, Telex 12 563

Rumänien
Siemens Birou
de consultatii tehnice
Strada Edgar Quinet 1
R-7 Bucuresti 1
Tel. 15 18 25, Telex 11 473

Schweden
Siemens AB
Norra Stationsgatan 63-65
Stockholm
(Fack, S-10435 Stockholm 23)
Tel. 22 96 80, Telex 1880/81

Schweiz
Siemens-Albis AG
CH-8001 Zürich
Löwenstraße 35
(CH-8021 Zürich,
Postfach 605)
Tel. 23 03 52, Telex 52 131

Spanien
Siemens S.A.
Orense, 2
Madrid-20 (Apartado 155)
Tel. 4 55 25 00, Telex 27 769

Tschechoslowakei
EFEKTIM
Vertretung ausländischer
Gesellschaften in der ČSSR
Václavské náměstí 1
CS-11000 Praha 1
(P.O.B. 457)
Tel. 25 84 17, Telex 122 389

Türkei
Simko Ticaret ve Sanayi A.S.
Meclisi Mebusan Cad. 55/35
Istanbul (Findıklı)
(P.K. 64 Tophane)
Tel. 45 20 90, Telex 22 290

Ungarn
INTERCOOPERATION Rt.
Siemens-Kooperations-
büro
Böszörményi út 9-11
H-1126 Budapest
(P.O.B. 1525)
Tel. 15 49 70, Telex 224 133

Union der Sozialistischen Sowjetrepubliken
Ständige Vertretung der
Siemens AG in Moskau
Internationales Postamt
Postfach 77
SU-Moskau
Tel. 2 23 52 57, Telex 7413

Afrika

Ägypten
Siemens Resident Engineers
P.O.B. 775, Zamalek
Cairo/Egypt
Tel. 3 56 61, Telex 321

Algerien
Siemens Algérie S.A.R.L.
3, Viaduc du Duc des Cars
Alger (B.P. 224, Alger-Gare)
Tel. 63 95 47, Telex 52 817

Äthiopien
Siemens Ethiopia Ltd.
Ras Bitwoded Makonen
Building
Addis Ababa (P.O.B. 5505)
Tel. 15 15 99, Telex 21052

Libyen
Assem Azzabi, Tariq Building
1, September Street
Tripoli (P.O.B. 2583)
Tel. 4 15 34 Telex 20029

Marokko
SETEL
Société Electrotechnique
et de Télécommunications S.A.
Rue Lafuente
Casablanca
Tel. 26 13 82/84, Telex 21914

Südafrika
Siemens (Proprietary) Limited
Siemens House
Corner Wolmarans and Biccard
Streets, Braamfontein
Johannesburg 2000
(P.O.B. 4583)
Tel. 7 25 25 00, Telx 587 721

Sudan
National Electrical
Commercial Company (NECC)
Khartoum (P.O.B. 1202)
Tel. 8 08 18, Telex 642

Tunesien
Sitelec S.A. Société
d'Importation
et de Travaux d'Electricité
26, Avenue Farhat Hached
Tunis
Tel. 24 28 60, Telex 12 326

Zaire
Siemens Zaire S.P.R.L.
1222, Avenue Tombalbaye,
Kinshasa 1 (B.P. 9897)
Tel. 2 26 08, Telex 377

Amerika

Argentinien
Siemens S.A.
Av. Presidente
Julio A. Roca 530
Buenos Aires
(Casilla Correo Central 1232)
Tel. 30 04 11, Telex 121 812

Bolivien
Sociedad Comercial
é Industrial Hansa Ltda.
La Paz (Cajón Postal 1402)
Tel. 5 44 25, Telex 5261

Brasilien
Siemens S.A.
Rua Cel. Bento Bicudo, 111
BR-05069 Sao Paulo
(Caixa Postal 1375),
Sao Paulo 1, SP)
Tel. 2 60 26 11, Telex 11-23681

Chile
Gildemeister S.A.C.
División Siemens
Casilla 99-D
Santiago de Chile
Tel. 8 25 23, Telex sgo 392

Kanada
Siemens Canada Limited
7300 Trans-Canada-Highway
Pointe Claire, P.Q. H9R 1C7
(P.O.B. 7300, Pointe Claire,
P.Q. H9R 4R6)
Tel. 695-7300, Telex 5 267 300

Kolumbien
Siemens S.A.
Carrera 65, No. 11–83
Bogotá
(Apartado Aéreo 80150)
Tel. 61 40 77, Telex 44 750

Mexiko
Siemens S.A.
Poniente 116, No. 590
Mexico 15, D.F.
(Apartado Postal 15064)
Tel. 5 67 07 22, Telex 17 72 700

Uruguay
Conatel S.A.
Ejido 1690
Montevideo
(Casilla de Correo 1371)
Tel. 91 73 31, Telex 934

Venezuela
Siemens S.A.
Apartado 3616
Caracas 101
Tel. 34 85 31, Telex 25 131

Vereinigte Staaten von Amerika
Siemens Corporation
186 Wood Avenue South
Iselin, New Jersey 08830
Tel. 4 94-1000
Telex WU 84-4491, 84-4492

Asien

Afghanistan
Siemens Afghanistan Ltd.
Alaudin, Karte 3
Kabul (P.O.B. 7)
Tel. 4 14 60

VX

Bangladesh
Siemens Bangladesh Ltd.
74, Dilkusha Commercial Area
Dacca (P.O.B. 33)
Tel. 24 43 81, Telex 824

Burma
Siemens Resident Engineer
8 Attia Road
Rangoon (P.O.B. 1427)
Tel. 3 25 08, Telex 2009

Hongkong
Jebsen & Co., Ltd.
Prince's Building, 23rd floor
Hong Kong (P.O.B. 97)
Tel. 5 22 5111, Telex 73221

Indien
Siemens India Ltd.
Head Office
134-A, Dr. Annie Besant Road,
Worli
Bombay 400018 (P.O.B. 6597)
Tel. 37 99 06, Telex 112 373

Indonesien
P.T. Siemens Indonesia
Kebon Sirih 4
Jakarta (P.O.B. 2469)
Tel. 5 10 51, Telex 46 222

Irak
Samhiry Bros. Co. (W.L.L.)
Abu Nawas Street
Baghdad (P.O.B. 300)
Tel. 9 00 21, Telex 2 255

Iran
Siemens Sherkate
Sahami (Khass)
Kh. Takhte-Djamshid 32
Siemenshaus
Teheran 15
Tel. 6141, Telex 212 351

Israel
Transelectro Company Ltd.
72/76 Harakevet Street
Tel Aviv (P.O.B. 2385)
Tel. 3 18 44, Telex 33 513

Japan
Nippon Siemens K.K.
Furukawa Sogo Building,
6-1, Marunouchi, 2-chome
Chiyoda-ku
Tokyo 100
(Central P.O. Box 1144
Tokyo 100-91)
Tel. 2 14 02 11, Telex 22 808

Jemen
Tihama Tractors
& Engineering Co. Ltd.
Sana'a (P.O.B. 49)
Tel. 24 62, Telex 217

Korea (Republic)
Siemens Electrical
Engineering Co., Ltd.
C.P.O. Box 3001
Seoul
Tel. 24 15 58, Telex 2329

Kuwait
Abdul Aziz M. T. Alghanim Co.
& Partners
Kuwait, Arabia (P.O.B. 3204)
Tel. 42 33 36, Telex 2 131

Libanon
Ets. F.A. Kettaneh S.A.
(Kettaneh Frères)
Rue du Port
Beyrouth (P.O.B. 110242)
Tel. 22 11 80, Telex 20 614

Malaysia
Guthrie Eng. (Malaysia) Sdn. Bhd.
Electrical & Communications
Division
17, Jalan Semangat
Petaling Jaya/Selangor
(P.O.B. 30)
Tel. 77 33 44, Telex 37 573

Pakistan
Siemens Pakistan
Engineering Co. Ltd.
ILACO House,
Abdullah Haroon Road
Karachi (P.O.B. 7158, Karachi 3)
Tel. 51 60 61, Telex 820

Philippinen
Engineering Equipment, Inc.
Machinery Division,
Siemens Department
P.O.B. 7160 Airmail Exchange Office
Manila International Airport
Philippines 3120
Tel. 85 40 11/19, Telex EEC 3695

Saudi-Arabien
E.A. Juffali & Bros.
Head Office
Jeddah (P.O.B. 1049)
Tel. 2 22 22, Telex 40 130

Singapur
Guthrie Engineering (Singapore)
Pte. Ltd.
Electrical
& Communications Division
41, Sixth Avenue,
Bukit Timah Road
Singapore 10
(P.O.B. 495, Singapore 1)
Tel. 66 25 55, Telex 21 681

Syrien
Syrian Import
Export & Distribution
Co., S.A.S. SIEDCO
Port Said Street
Damas (P.O.B. 363)
Tel. 1 34 31/33

Taiwan
Delta Engineering Ltd.
42, Hsu Chang Street,
8th floor
Taipei (P.O.B. 58497)
Tel. 3 61 02 55, Telex 21 826

Thailand
B. Grimm & Co. R.O.P.
1643/4, Petchburi Road
(Extension)
Bangkok 10 (P.O.B. 66)
Tel. 52 40 81, Telex 2614

Australien und Ozeanien

Australien
Siemens Industries Ltd.
544 Church Street, Richmond
Melbourne, Victoria 3121
Tel. 4 29 7111, Telex 30 425

Neuseeland
Siemens Liaison Office
175 The Terrace
Wellington 1 (P.O.Box 4145,
G.P.O.
Tel. 72 98 61, Telex 31233

Inhaltsverzeichnis

Einführung

Zusammenbau

Erklärungen zur Arbeitsweise
Ihres Mikrocomputers

Endzusammenbau und Überprüfung

Das MONITOR-Programm

Ausdruck des MONITOR-Primärprogrammes

Gesamtschaltung
des SIKIT-DK/8080-Computers

Gesamtbestückungsplan

Anschriften unserer Geschäftsstellen

blank page

SIEMENS

Bestell-Nr. B 1578
Printed in West Germany
KG 7761.

VX